百位专家 十年巨献 院士推荐

全面学习●深入培训●实战技巧●项目经验●案例代码●商用资源

您的私人编程专家

学习更轻松，开发更方便！

⊙ 一本图书的价格 ⊙ 二十本书的容量 ⊙ 传统书不可比拟的功能 ⊙

C#

编程全能词典

BIANCHENGQUANNENGCIDIAN

明日科技 编著



软件 + 图书

编程词典（双DVD） 快速查询手册

15000 页技术文档	2788 个实例应用	21 个实用项目	30小时 开发录像
515 个典型范例	2940 多个设计界面	120 多个开发方案	1年 12次升级

Broadview® www.broadview.com.cn

编程全能词典技术服务与销售热线：
400-675-1066 / 0431-84978981 84978982
网址：www.mrbccd.com

明日科技 MINGRISOFT

编程全能词典系列产品

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

更多资源请访问稀酷客(www.ckook.com)

Broadview®
www.broadview.com.cn
博文视点·IT出版旗舰品牌
技术凝聚实力·专业创新出版

十年功夫汇聚成《编程全能词典》，这份心血令人钦佩。

——中国工程院院士 王家耀

用户对象：

◎零基础者

28堂课、30小时学习视频、437个实战训练任务、56个快乐编程驿站程序、数位专家在线支持、没有基础也能轻松学会编程。

◎开发人员

技术资源库+实例资源库+项目资源库+方案资源库+界面资源库+源码资源库+工具资源库+辅助词典库+个人资源库+全能查询，提高开发效率100%。

◎项目经理

本产品开发团队凭借多年开发经验，提供了经过商业实践过的项目方案，界面方案与技术方案，让您完成项目更轻松。



编程词典诞生大事记：

◆2000年5月，明日科技团队将为编程者提供有价值内容和产品（编程词典）作为公司发展重点之一。

◆2003年1月，明日科技第一本编程图书问世，获得了广大读者的支持和鼓励，在同类图书中长期排名第一。

◆2005年8月，明日读者服务论坛和支持网站（www.mingribook.com）正式运行，目前成为国内图书服务领域的品牌。

◆2007年9月，明日科技第一本图书版权输出到台湾，目前已有近20本图书版权输出到台湾。

◆2009年11月，明日编程类图书已形成软件工程师入门和软件工程师典藏两大类，共200多本图书，销售码洋超过13000万。

◆2010年1月，编程词典系列产品在北京图书订货会上正式发布，同期编程词典全程服务社区www.mrbccd.com正式发布。

**请务必严格遵循光盘说明和书中“快速安装指南”安装使用本产品。

上架建议：程序开发>C#

ISBN 978-7-121-09833-8
9 787121 098338 >
定价：98.00元（含DVD光盘2张）

责任编辑：高洪霞
责任美编：李 玲

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书。

更多资源请访问稀酷客(www.ckook.com)

编程全能词典系列产品

C#
编程全能词典

BIANCHENGQUANNENGCIDIAN

明日科技 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

TP3/25
M807

更多资源请访问稀酷客(www.ckook.com)

内 容 简 介

《C#编程全能词典》是专门为广大程序设计人员开发的高效编程学习产品；使用方便、快捷，易操作，上手快。它是读者学习的老师、技术查询的工具、互动与交流的朋友。本书内容涵盖了技术、项目、方案、源代码、视频、界面等各个方面，既能学习编程知识，又能查询编程技术，帮读者快速解决学习和开发中遇到的各种难题。

无论是零基础的初学者、编程爱好者、培训机构的老师和学员，还是已经从事编程工作多年的开发人员，编程全能词典都同样适合阅读和参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

C#编程全能词典 / 明日科技编著. —北京：电子工业出版社，2010.3
（编程全能词典系列产品）
ISBN 978-7-121-09833-8

I. C… Ⅱ. 明… Ⅲ. C 语言－程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2009）第 202186 号

责任编辑：高洪霞
印　　刷：涿州市京南印刷厂
装　　订：涿州市桃园装订有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：900×1280　1/32　印张：16.125　字数：564 千字
印　　次：2010 年 3 月第 1 次印刷
印　　数：4000 册　定价：98.00 元（含 DVD 光盘 2 张）

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

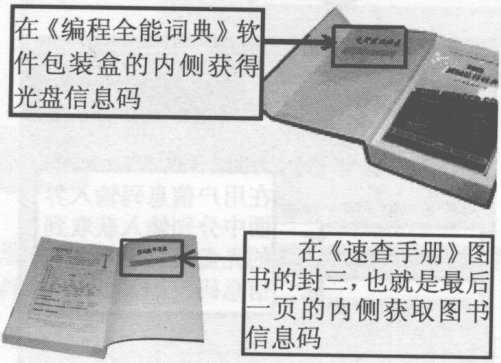
服务热线：（010）88258888。

更多资源请访问稀酷客(www.ckook.com)

快速安装指南

1 获取用户信息码

本产品是结合图书销售的，用户在安装前，需获得本软件的图书信息码和光盘信息码。图书信息码在《编程全能词典》图书的封 3 可以获得，光盘信息码在包装盒的内侧盒顶可以获得。



图书信息码为 12 位数字，示例如下图：

753867264531

光盘信息码为 12 位数字，示例如下图：

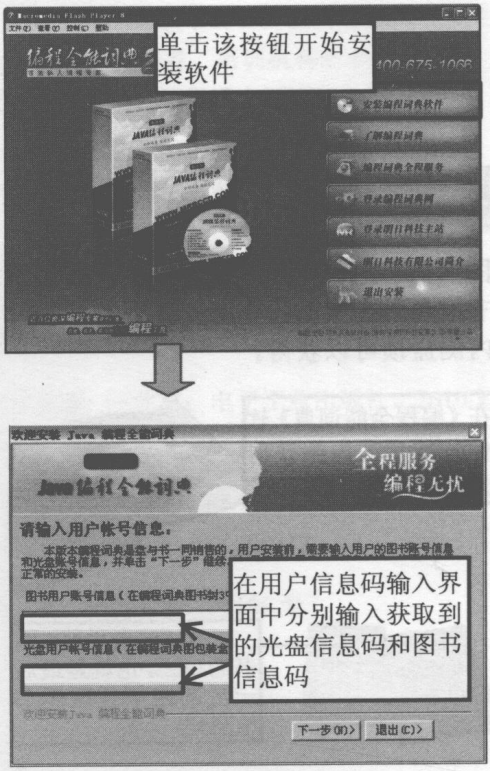
609633263819

请妥善保留信息码。

更多资源请访问稀酷客(www.ckook.com)

2 安装编程词典产品

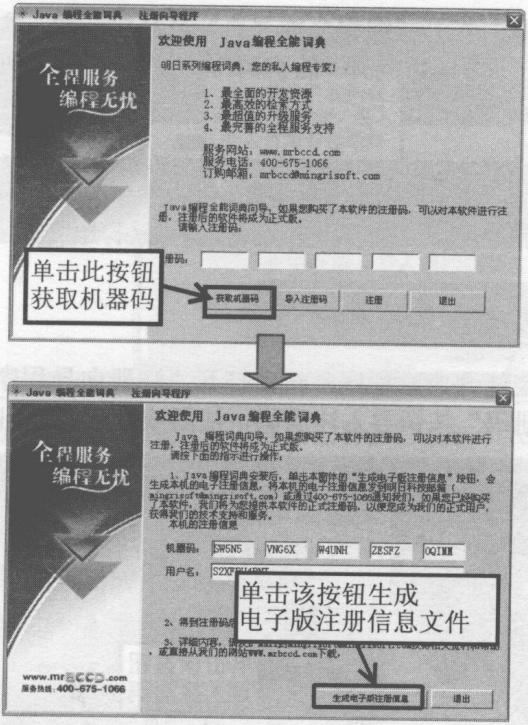
将《编程全能词典》产品光盘放入光驱后，进入“安装导航”界面，单击“安装编程全能词典软件”按钮，进入输入用户信息码界面，在该界面中分别输入已经获取到的光盘信息码和图书信息码。然后单击“下一步”按钮即可进行安装。



3 生成电子版注册信息

安装成功后，运行《编程全能词典》产品，将会打开“注册向导程序”窗口。分别单击“获取机器码”和“生成电子版注册信息”按钮生成电子版注册信息。

更多资源请访问稀酷客(www.ckook.com)

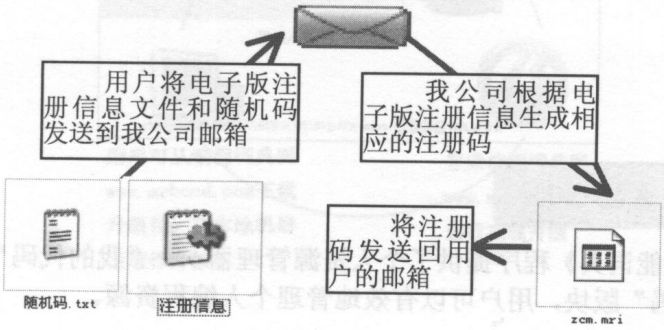


4 获取注册码

电子版的注册信息生成以后，将弹出提示对话框，并在对话框中显示电子版注册信息文件的保存路径。

用户要将生成的电子版注册信息文件和随机码通过 E-mail 发送到我公司的邮箱中，我公司的邮箱地址为：mingrisoft@mingrisoft.com。

我公司在收到用户的注册信息和随机码以后，会生成相应的注册码并通过 E-mail 发送到用户的邮箱中。用户收到注册码并保存到《编程全能词典》产品的安装路径以后，就可以对产品进行注册了。



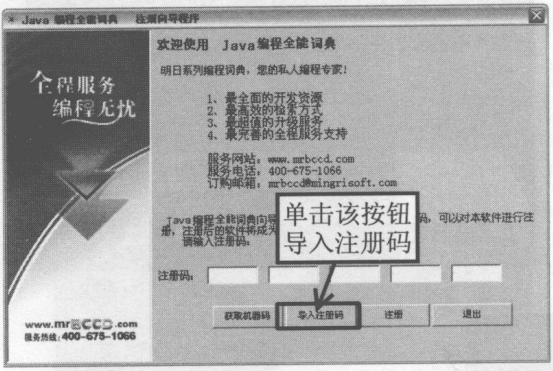
更多资源请访问稀酷客(www.ckook.com)

“随机码”位置：打开《编程全能词典》产品包装盒，拿出光盘盒，在存放光盘盒位置的正下方即可刮开“随机码”，用户需将随机码的信息以文本文件的形式进行保存。

> 注意：发送至我公司的电子版注册信息文件一定是电脑自动生成的文件，避免手动输入。

5 注册并使用编程词典软件

运行《编程全能词典》程序，将会打开“注册向导程序”窗口。在窗口中单击“导入注册码”按钮导入注册码，注册成功后会弹出提示对话框。



> 注意：用户一定要通过单击“导入注册码”按钮来导入注册码，避免手动输入。

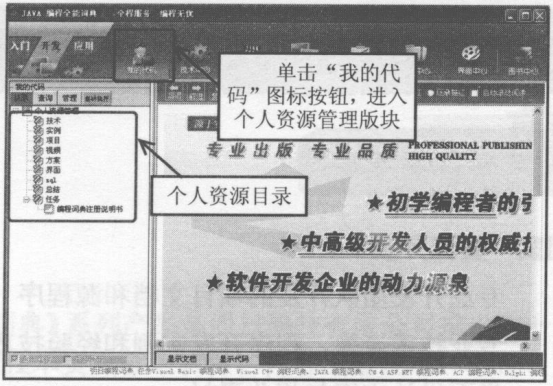
在提示对话框中单击“确定”按钮，即可运行并使用《编程全能词典》产品。



6 个人资源管理

《编程全能词典》程序提供了个人资源管理器——“我的代码”版块，通过“我的代码”版块，用户可以有效地管理个人编程资源。

更多资源请访问稀酷客(www.ckook.com)



在“我的代码”版块中，用户可以根据需要自行添加个人编程资源，比如：

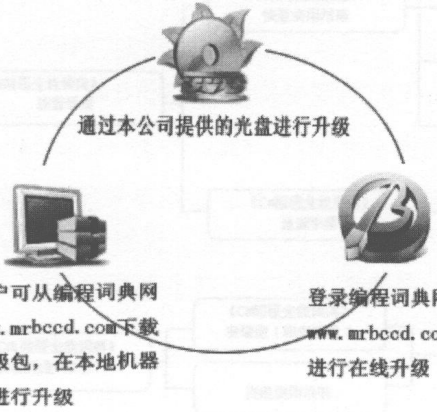
- 添加编程资源的类别（类别可修改和删除）
- 添加个人的编程资源
- 修改和删除已添加的个人编程资源
- 浏览已添加的个人编程资源

注意：编程资源的类别可以多样化，如技术、实例、项目等，添加的编程资源也可以是文档、图片、代码等多种形式。

7 有限技术支持

购买《编程全能词典》的用户，可以享受一年 12 次免费产品内容升级。具体升级内容可登录编程词典网 www.mrbccd.com 进行查看。

以下是《编程全能词典》产品的三种升级方式：



更多资源请访问稀酷客(www.ckook.com)

注意：通过光盘进行升级的编程词典用户，本公司将收取光盘的成本费及邮递费用。

8 更多升级和服务

专业开发团队　资源升级

- 专业开发团队开发的项目文档和源程序
- 专业技术录像，程序开发实例和经验技巧
- 界面设计方案与相关素材
- 不断升级的编程实用工具

专业技术支持　全程服务

- 服务团队来自开发一线，实战经验丰富
- 提供全程专业服务，全面解决开发问题
- 四位一体服务

400 电话支持：400-675-1066

网站论坛支持：bbs.mrbccd.com

QQ 在线服务：200958602（C#）

TQ 在线服务：登录编程词典网，点击即可对话

更多资源请访问稀酷客(www.ckook.com)

前　言

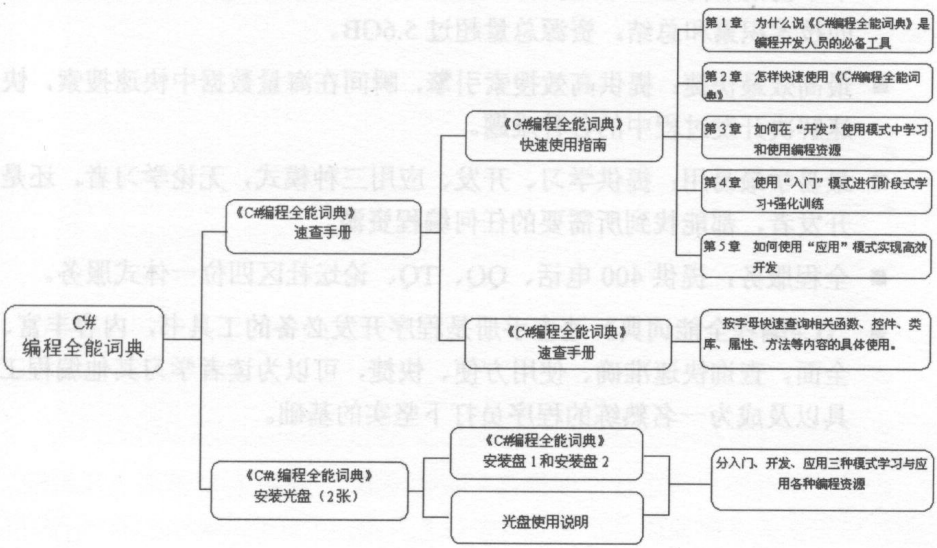
《编程全能词典》系列产品是明日科技有限公司推出的一款各级编程人员不可缺少的编程技术资源管理软件。内容涵盖技术、项目、方案、源码、视频、界面等方面。是您学习的老师、技术查询的工具、互动与交流的朋友。

对于学习者，它就像您的私人老师，分阶段指导您学习编程，并结合训练题加以巩固；对于开发者，它就是您的资源宝典和私人开发团队，既能查询最新、最权威的编程技术，又能协助您编程，解决您开发中遇到的各种编程难题。

365 天，随时都能让您学习并使用到最新、最权威的编程技术，随时都能让您享受到最快捷、最周到的售后服务。

本书内容：

本书提供了一本《C#编程全能词典》速查手册和两张《C#编程全能词典》的光盘。大体结构如下图所示。



更多资源请访问稀酷客(www.ckook.com)

《C#编程全能词典》速查手册：

《C#编程全能词典》速查手册包括《C#编程全能词典》快速使用指南和《C#编程全能词典》速查手册两方面内容。

《C#编程全能词典》快速使用指南分 5 章，全面、系统地介绍了编程全能词典内容、如何快速使用、如何分模式使用编程全能词典相关编程资源；《C#编程全能词典》速查手册是一本工具书，示例丰富、栏目精彩、典型应用贴近实际，包括了相关语言的所有控件、函数、属性、类等各种技术，按字母进行排序，读者可以更快、更准确地查找相关内容。

《C#编程全能词典》安装光盘：

《C#编程全能词典》产品提供 2 张安装光盘，同时提供一个“光盘使用说明”文档，读者可根据光盘使用说明快速安装及注册《C#编程全能词典》产品。同时通过“光盘使用说明”文档中提供的“编程全能词典快速使用导航”了解《C#编程全能词典》的使用。

《C#编程全能词典》特点

- 最权威：汇聚明日科技数百位开发人员近 6 年的心血。
- 最全面最系统：目前国内最系统、最全面的开发资源，涵盖明日科技公司开发的所有项目文档和图书资源，同时还包含明日科技出版的数十本畅销书内容，绝无仅有。汇聚明日科技数十位程序开发人员多年的技术积累和总结，资源总量超过 5.6GB。
- 最高效最快捷：提供高效搜索引擎，瞬间在海量数据中快速搜索，快速解决开发过程中的各种难题。
- 最易学最易用：提供学习、开发、应用三种模式，无论学习者，还是开发者，都能找到所需要的任何编程资源。
- 全程服务：提供 400 电话、QQ、TQ、论坛社区四位一体式服务。
- 《C#编程全能词典》速查手册是程序开发必备的工具书，内容丰富、全面，查询快速准确、使用方便、快捷，可以为读者学习其他编程工具以及成为一名熟练的程序员打下坚实的基础。

更多资源请访问稀酷客(www.ckook.com)

《C#编程全能词典》读者对象

- ☑ 初学编程的自学者
- ☑ 编程爱好者
- ☑ 大中专院校的老师和学生
- ☑ 相关培训机构的老师和学员
- ☑ 初、中级程序开发人员
- ☑ 程序测试及维护人员
- ☑ 参加实习的“菜鸟”程序员

读者服务

为方便读者，明日科技有限公司提供了 400 电话、QQ、TQ、论坛社区等 4 位一体式服务。

- 400 电话：400-675-1066
- QQ：200958602（C#）
- TQ：C#编程词典
- 论坛：bbs.mrbccd.com

读者可通过上述四位一体式服务联系我们，我们力求在早 8:00—晚 21:00 点（周六、周日除外）的 13 小时内提供技术支持。

更多资源请访问稀酷客(www.ckook.com)

光盘使用说明

欢迎使用《C#编程全能词典》产品，希望它能成为您编程开发的左膀右臂、私人编程专家。

在使用《C#编程全能词典》产品之前，需要将《C#编程全能词典》产品进行安装、注册等操作，具体操作流程如下：



图 1　光盘使用流程图

快速安装《C#编程全能词典》

为使您更好地使用《C#编程全能词典》产品，你的计算机需要按如下的软硬件进行配置：

- 操作系统：Windows XP / 2000 / 2003 / Vista
- CPU：Pentium 3 以上
- 内存：256MB 以上
- 硬盘空间：8GB 以上

安装《C#编程全能词典》产品的具体操作步骤如下：

1 将《C#编程全能词典》安装光盘放入光驱，光盘将自动运行，并弹出“欢迎安装”界面。

如果光盘不能自动运行，可打开“我的电脑”，鼠标右键单击光盘图标，在弹出的快捷菜单中选择“打开”命令打开光盘，双击 index.exe 文件，也可以进入“欢迎安装《C#编程全能词典》”界面，如图 2 所示。

更多资源请访问稀酷客(www.ckook.com)

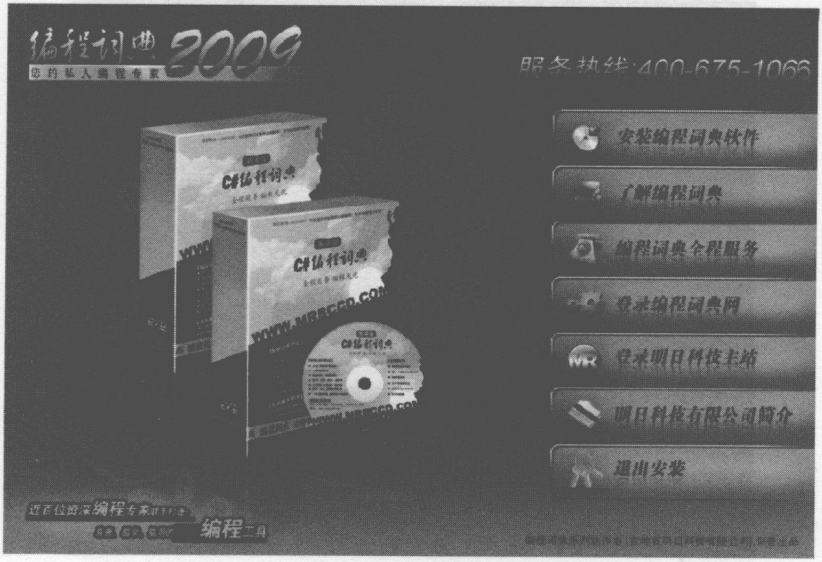


图 2　安装选择界面

② 单击“安装编程词典软件”图标按钮，进入“欢迎安装 C#编程全能词典”界面，按提示输入“图书用户账号信息”及“光盘用户账号信息”。如图 3 所示。

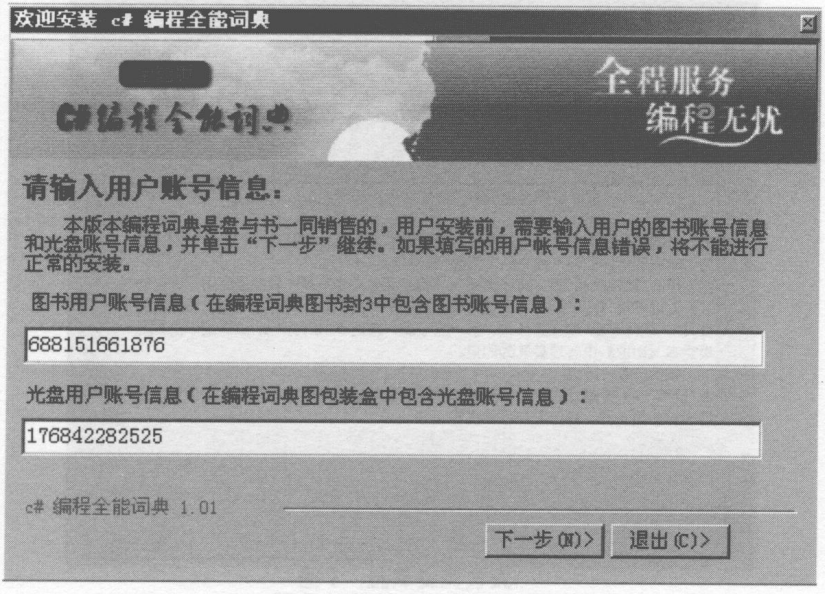


图 3　欢迎安装界面

③ 单击“下一步”按钮，进入 C#编程全能词典安装界面，如图 4 所示。

更多资源请访问稀酷客(www.ckook.com)

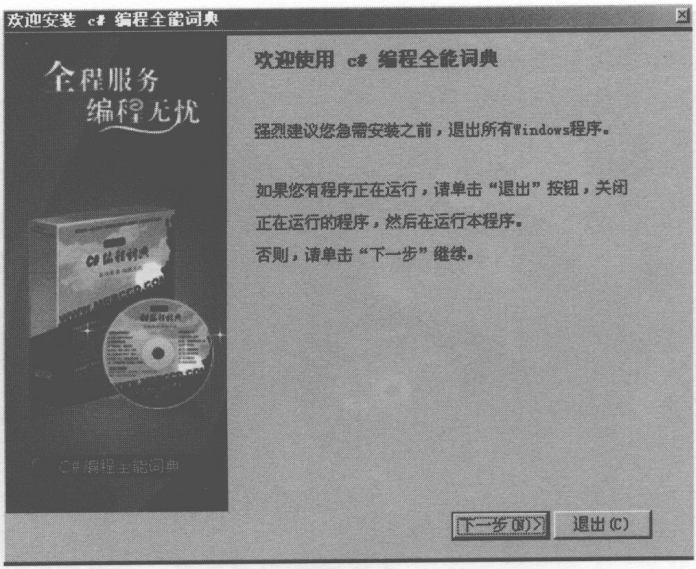


图 4　C#编程全能词典安装界面

4 单击“下一步”按钮，进入“许可证协议”界面，如图 5 所示。认真阅读许可协议，然后选择“我同意该许可协议的条款”单选按钮，以便继续进行安装。

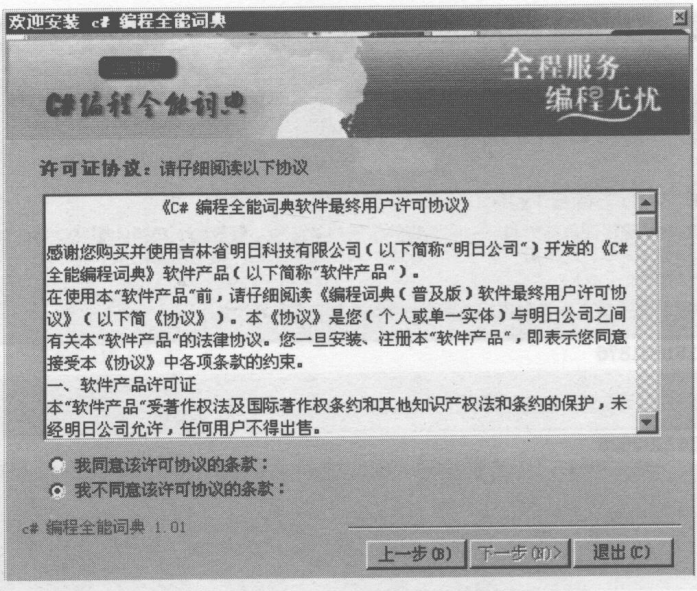


图 5 《许可证协议》界面

5 单击“下一步”按钮，进入“用户信息”界面，如图 6 所示。在“用户”和“公司”文本框中分别输入相应的内容。

更多资源请访问稀酷客(www.ckook.com)

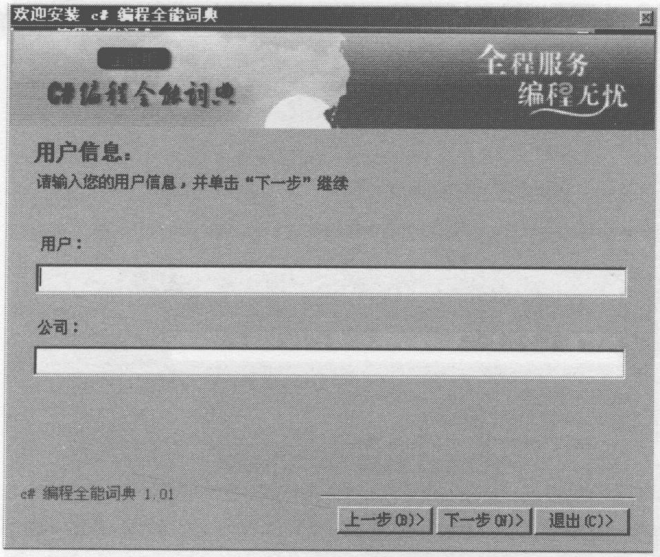


图 6　用户信息界面

6 单击“下一步”按钮，进入“选择安装方式”界面，如图 7 所示。在该界面用户可以选择“全部安装”或“部分安装”。这里选择“全部安装”方式，所以勾选“全部安装”复选框。

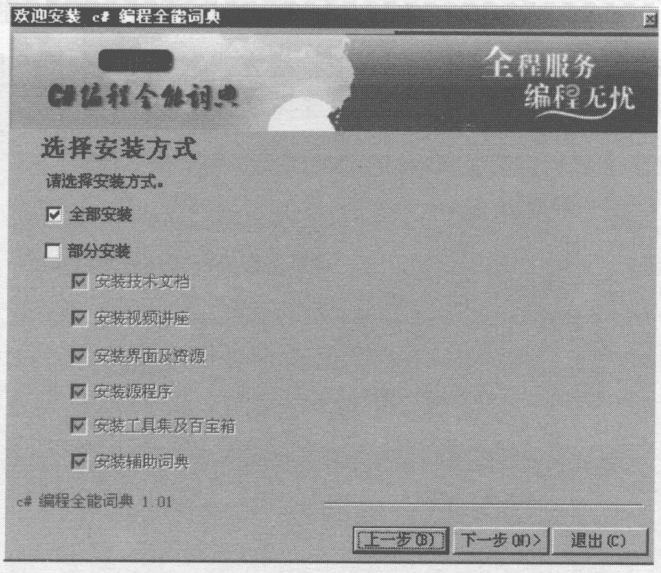


图 7　选择安装方式

7 单击“下一步”按钮进入“安装文件夹”界面，如图 8 所示。在该界面用户可以选择默认安装路径，也可以通过“更改”按钮更改安装路径，这里选择默认的安装路径。

更多资源请访问稀酷客(www.ckook.com)

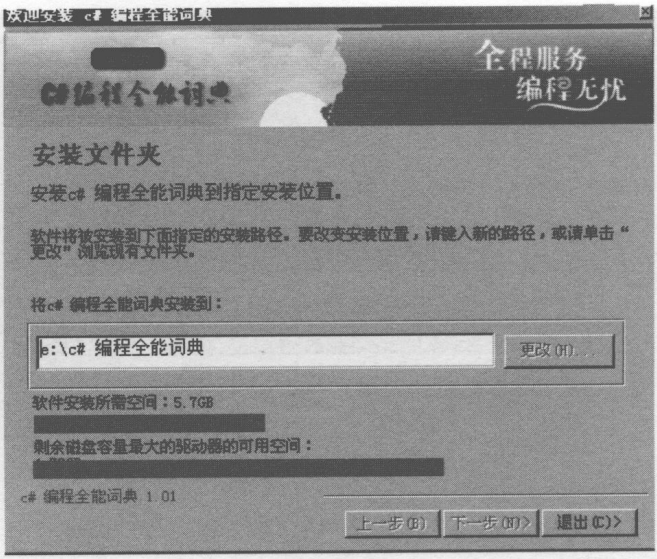


图 8 “安装文件夹”界面



注意

选择安装路径时必须确保安装路径有足够的空间。

8 单击“下一步”按钮，进入“快捷方式”界面，如图 9 所示。用户可以根据实际需要选择快捷方式，这里选择默认设置。

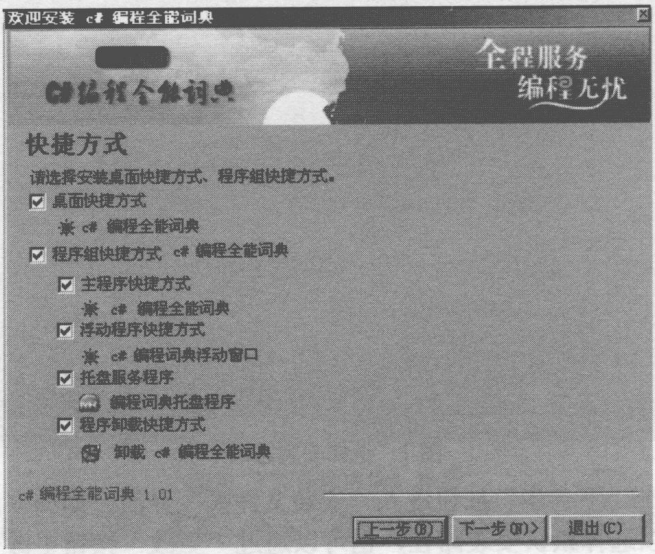


图 9 “快捷方式”界面

更多资源请访问稀酷客(www.ckook.com)

⑨ 单击“下一步”按钮，进入“准备安装”界面，如图 10 所示。

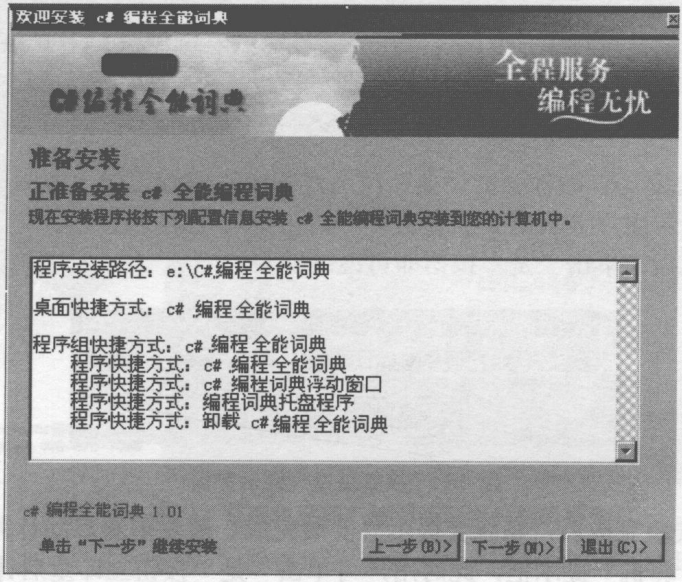


图 10 “准备安装”界面

⑩ 继续单击“下一步”按钮，进入“安装”界面，单击“开始安装”按钮即可进行安装，如图 11 所示。

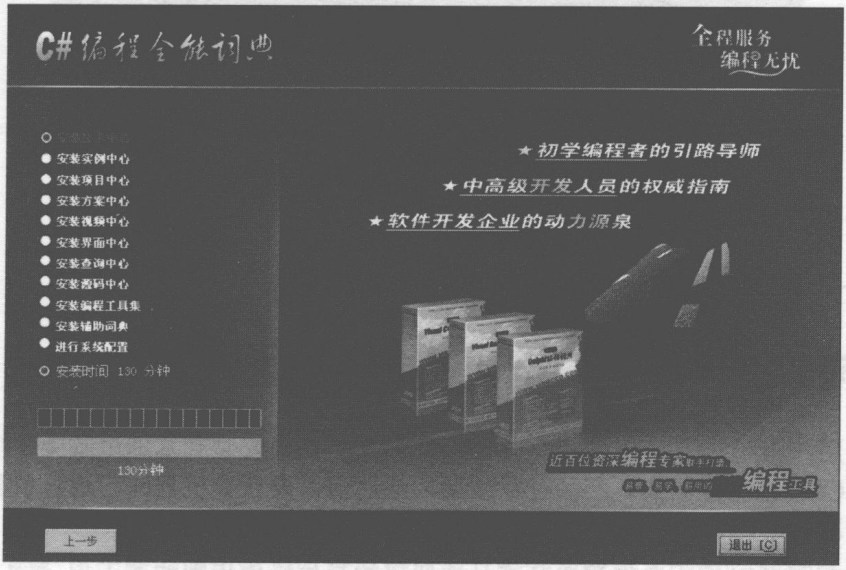


图 11 “安装”界面

更多资源请访问稀酷客(www.ckook.com)

提示

在安装过程中，用户可以在安装界面的右侧浏览关于C#编程全能词典的相关信息，如各版本的不同功能及不同版块中所包含的内容等。

11 安装《C#编程全能词典》产品过程中，系统会弹出如图 12 所示的提示框，提示用户放入第 2 张光盘。此时用户可将第 1 张光盘取出，然后放入第 2 张光盘，单击“是”按钮即可继续安装。

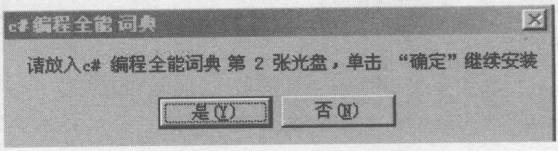


图 12 “换安装盘”提示界面

12 安装完成后，系统会提示“安装完成，需要重新启动计算机，是否立即重启计算机”提示框，此时用户可单击“是”按钮立即重启计算机。

13 计算机重启之后，会自动弹出如图 13 所示的程序更新的界面。



图 13 “程序更新”界面

如何获取《C#编程全能词典》注册码

用户将购买的《C#编程全能词典》产品成功地安装到本地电脑中，需要对《C#编程全能词典》产品进行注册，注册前需要获取机器码，通过机器码来获取注册码。

获取《C#编程全能词典》产品注册码的具体操作步骤如下：

1 单击“开始”按钮，在“开始”菜单中选择“所有程序”/“C#编程全能词典”命令，打开“C#编程全能词典 注册向导程序”窗口，在窗口中单

更多资源请访问稀酷客(www.ckook.com)

击“获取机器码”按钮获取用户的机器码，如图 14 所示。

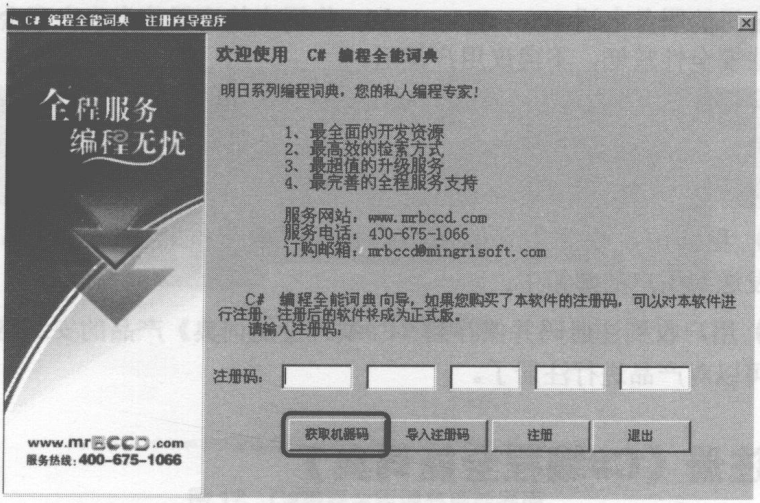


图 14　通过“获取机器码”按钮获取用户的机器码

2 单击“获取机器码”按钮后，注册向导程序将自动获取用户的机器码，并显示在注册向导窗口中，如图 15 所示。单击“生成电子版注册信息”按钮，将注册向导窗口中显示的“机器码”和“用户名”的注册信息生成为电子版的注册信息。

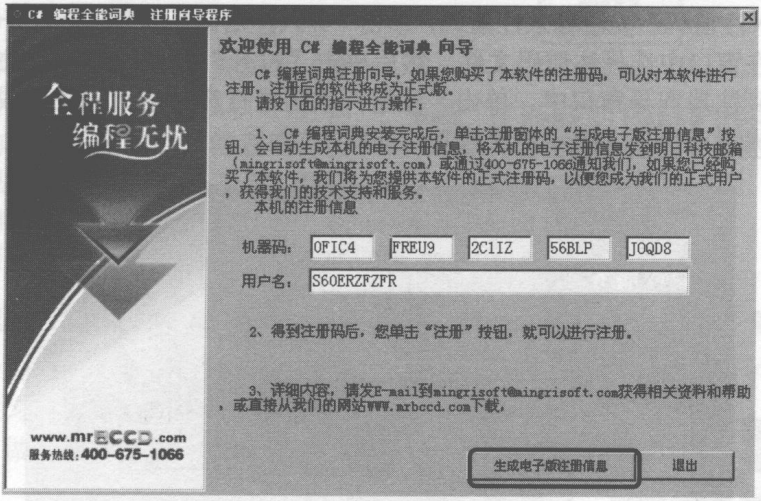


图 15　生成电子版注册信息

3 电子版的注册信息生成成功以后，将弹出提示对话框，并在对话框中显示电子版注册信息文件的保存路径。

4 用户需要将电子版的注册信息文件和随机码通过 E-mail 发送到我公

更多资源请访问稀酷客(www.ckook.com)

司的邮箱中，我公司的邮箱地址为：mingrisoft@mingrisoft.com。用户也可以通过我公司的服务电话400—675—1066，将用户的注册信息告之我公司，但此种方法安全性较低，不建议用户使用。

“随机码”位置：打开《编程全能词典》产品包装盒，拿出光盘盒，在存放光盘盒位置的正下方即可刮开“随机码”，用户需将随机码的信息以文本文件的形式进行保存。

5 我公司在收到用户的注册信息后，会生成相应的注册码并通过E-mail发送到用户的邮箱中。

6 用户收到注册码并保存到《C#编程全能词典》产品的安装路径中以后，就可以对产品进行注册了。

轻松注册《C#编程全能词典》

用户将获取的注册码复制到本地计算机中，然后对《C#编程全能词典》产品进行注册，注册成功后，《C#编程全能词典》产品就可以使用了。

注册《C#编程全能词典》产品的具体操作步骤如下：

1 单击“开始”按钮，在“开始”菜单中选择“所有程序”/“C#编程全能词典”命令，打开“C#编程全能词典 注册向导程序”窗口。在窗口中单击“导入注册码”按钮导入注册码，单击“导入注册码”按钮后会打开“打开”窗口。在窗口中选择注册码文件，然后单击“打开”按钮，即可将注册码自动添加到注册向导窗口中。单击“注册”按钮进行产品注册。注册成功后，将弹出提示对话框，提示注册成功。如图16所示。

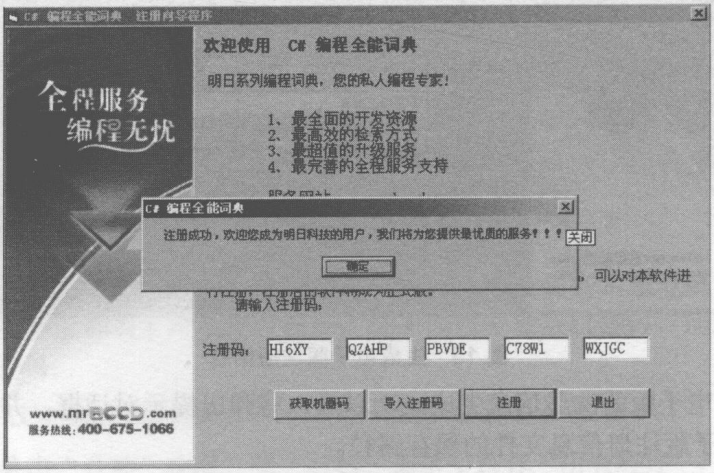


图16 导入注册码

更多资源请访问稀酷客(www.ckook.com)

② 在提示对话框中单击“确定”按钮，即可运行并使用《C#编程全能词典》产品，如图 17 所示。

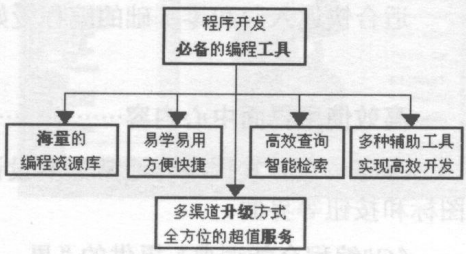


图 17　C#编程全能词典启动界面

《C#编程全能词典》快速使用导航

程序开发人员为什么使用《C#编程全能词典》 …… 第 1 章

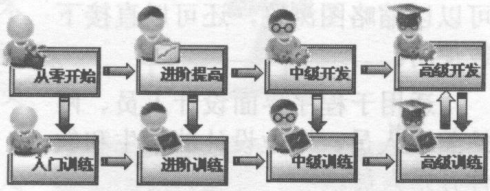
《C#编程全能词典》是程序开发人员必备的编程工具。



内容涉及海量的编程资源库（技术、实例、方案、界面、视频等）；独具特色的使用模式和高效的查询与检索；实现高效开发的实用工具、源码与各种辅助词典；多渠道的升级方式与全方面的超值服务。

快速学习编程技术（阶段学习+强化训练） …… 第 4 章

初学编程的人员最好的学习方式就是在学完一个知识点、一个例子之后，能通过练习的方式对所学的知识重新回顾，做到完全掌握，并能够举一反三。



《C#编程全能词典》提供了“入门”学习模式。它包括了 **335** 篇技术文档、**266** 套练习题，分入门、进阶、中级和高级 4 个阶段。通过学习、练习的方式来强化训练，加深对知识的理解和消化。

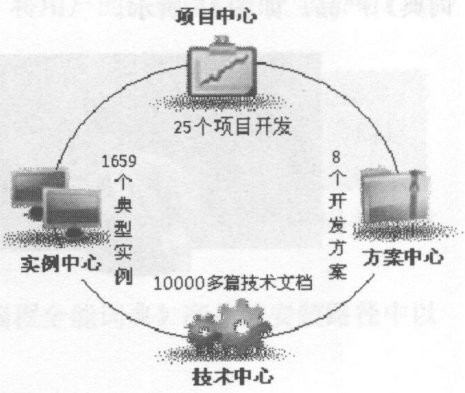
更多资源请访问稀酷客(www.ckook.com)

分类使用开发资源……第 3 章

技巧、实例、项目和方案不但可以解决开发中的常见问题，更可以提高程序员的编程水平。

《C#编程全能词典》提供了 **1659** 个典型实例、**10000** 多篇技术文档、**25** 个项目的开发、**8** 个开发方案。快速帮助编程者掌握技术、技巧实例和项目的开发。

适合各级软件开发人员使用，也可供大中专院校的师生学习参考。

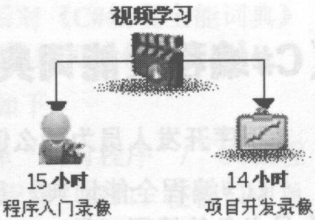


视频学习编程技术和项目开发……第 3 章 3.6 节

如果初学编程的爱好者想快速入门，那您可以到“视频中心”。

视频中心包括将近 **30** 个小时的视频录像。其中涉及程序设计入门的录像 **15** 小时，涉及项目开发的录像 **14** 小时。引导初学者快速入门。

适合快速入门和零基础的编程爱好者。

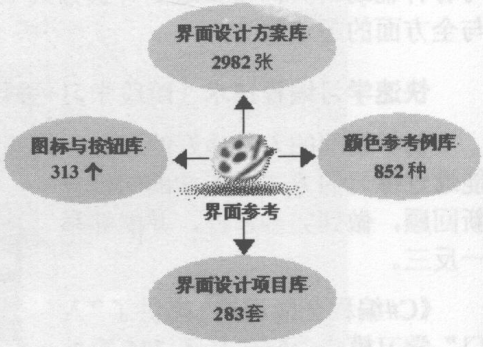


高效使用界面中心内容……第 3 章 3.7 节

程序员在开发程序时通常需要设计一些主界面、登录界面、查询界面、图标和按钮等界面。

《C#编程全能词典》提供的“界面中心”中包括了 **2982** 张界面设计方案、**313** 个图标和按钮、**283** 套项目界面套餐，以及 **852** 种颜色参考样例。编程者既可以分类浏览，也可以以缩略图浏览，还可以直接下载使用。

适用于程序界面设计人员、网站美工人员、毕业设计的学生和软件开发人员。



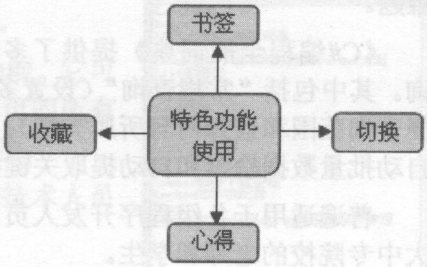
更多资源请访问稀酷客(www.ckook.com)

使用收藏,心得,书签 …… 第 2 章 2.3 节

编程人员在浏览文档或查询技术时通常会有一些想法或体会，浏览到好的文档时通常希望加以收藏。在不断积累编程知识的同时也可以提高编程水平。

《C#编程全能词典》提供了收藏、心得、书签、切换等一些特色功能。既可以隐藏目录大屏幕浏览文档并加以收藏，也可以记录一些心得和体会。

适用于初、中级程序开发人员、大中专院校的老师、学生，以及一些编程爱好者。



使用浮动窗口进行快速查询 …… 第 2 章 2.3.3 节

编程人员在编写文档或开发程序时通常需要在文档（或程序）和查询软件之间进行相互切换，比较麻烦。

《C#编程全能词典》提供了浮动查询窗口，用户在启动电脑时浮动查询窗口将自动运行，并隐藏于屏幕上方，不妨碍正常的工作。需要时将浮动窗口大化，即可快速查询相关技术。

普遍适用于初、中级程序开发人员和从事开发的技术人员。

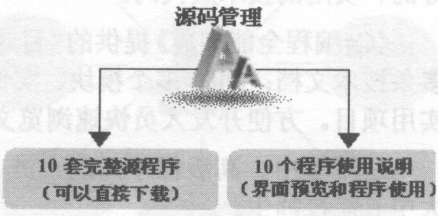


有效利用源码实现选择性下载 …… 第 5 章 5.3 节

在程序开发过程中，如果可以将某些程序的代码直接复制下来在开发环境中修改并使用，那么在一定程度上会加快二次开发的过程。

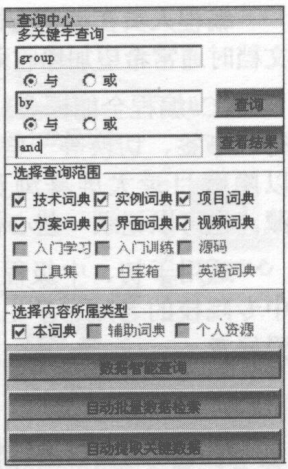
《C#编程全能词典》提供的“源码管理”包括 10 套完整的源程序和 10 个程序使用说明书。程序员可以通过阅读说明书（包括程序使用和界面预览）实现有选择地下载源程序。

适用于有一定语言基础的编程人员和软件开发人员。



更多资源请访问稀酷客(www.ckook.com)

智能高效查询编程技术……第 5 章 5.2 节

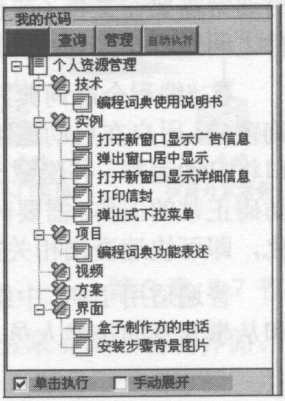


作为一名程序开发人员，通常需要通过多元化、高效的检索方式来快速解决开发过程中的各种技术难题。

《C#编程全能词典》提供了多种方式的高效查询。其中包括“常规查询”（设置多关键字查询、选择查询范围和选择内容所属类型）；数据智能查询；自动批量数据检索和自动提取关键数据。

普遍适用于各级程序开发人员、软件开发人员、大中专院校的老师和学生。

有效管理个人资源……第 3 章 3.1 节

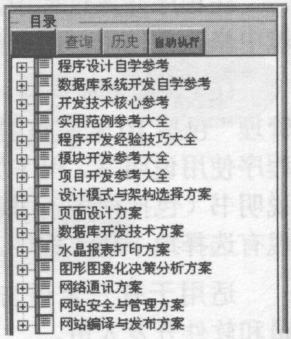


编程人员在编程过程中会积累一些技巧、实例、项目、方案和界面等个人资源。

《C#编程全能词典》提供的“我的代码”可以帮助编程人员管理这些个人资源。其中的“资源管理”可帮助编程人员对技巧实例、方案等资源进行分类管理（添加、修改、删除、批量导入数据）。“资源浏览”可对个人资源进行分类浏览与搜索。

适用于中级程序开发人员、大中专院校师生使用。

利用目录快速学习编程技术……第 5 章 5.1 节



开发人员在做项目或开发时需要阅读大量的同类图书，了解不同技术的应用。以便从中借鉴一些好的、实用的技术及代码。

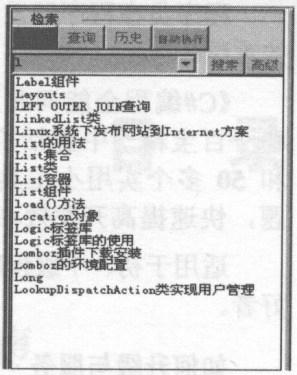
《C#编程全能词典》提供的“目录”包括了 **10000 多**条技术文档；**2000 多**个模块、实例、技巧；**25** 个实用项目。方便开发人员快速浏览文档。

适用于中、高级程序开发人员、编程爱好者和大中专院校师生。

更多资源请访问稀酷客(www.ckook.com)

利用个别关键字（字母）智能检索编程技术 …… 第 5 章 5.2.2 节

开发者在查询某些技术时如果能通过输入个别关键字或字母，迅速检索到包含该关键字的所有技术文档，则会大大提高程序员的开发效率。

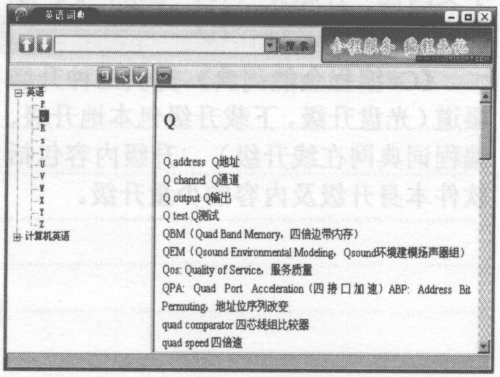


《C#编程全能词典》提供了“检索”功能。它可以通过输入单个字母快速检索到包含该字母的所有技术文档。大大提高了检索速度及检索深度。

适用于编程爱好者、从事开发的工程技术人员和中级程序开发人员。

使用英语辅助词典 …… 第 5 章 5.4 节

程序员在编程时如果手头的词汇量不够，会给程序开发带来一定的难度。

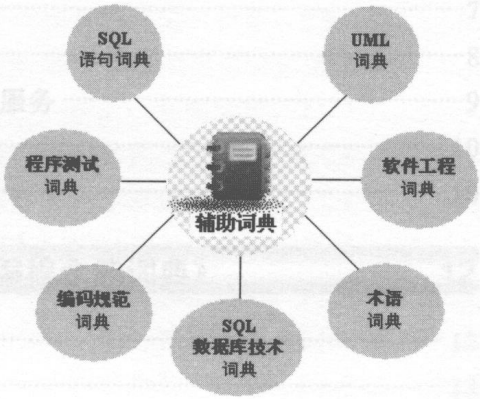


《C#编程全能词典》提供了“英语词典”，它收录了编程时使用频率最高的计算机英语和专业英语的汉语解释，程序员可随用随查，方便快捷。

适用于初、中级程序开发人员和编程爱好者。

使用其他辅助词典帮助编程 …… 第 5 章 5.5 节

在程序开发过程中程序员避免不了需要查询一些SQL语句、数据库知识、UML 语言、常用术语、编码规范等内容。



《C#编程全能词典》提供了 7 种辅助词典，即查即用，方便快捷。包括 SQL 语句词典、SQL 数据库技术词典、术语词典、UML 词典、编码规范词典、软件工程词典和程序测试词典。

适用于大中专院校的老师和学生、初、中级程序开发人员和做毕业设计的学生。

更多资源请访问稀酷客(www.ckook.com)

合理使用工具集与百宝箱 …… 第 5 章 5.6 节

程序员在程序开发过程中通常需要借助一些实用工具来解决开发中遇到的难题。

《C#编程全能词典》提供的“工具集”和“百宝箱”中囊括了 **20** 多个辅助工具和 **50** 多个实用小工具。轻松解决开发难题，快速提高开发效率。

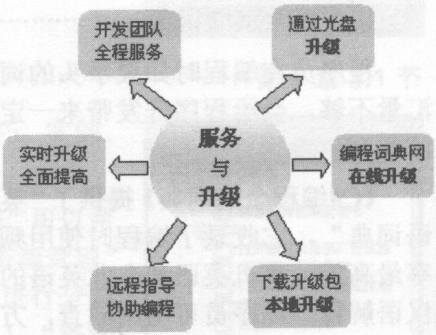
适用于初、中级程序开发人员和编程爱好者。



如何升级与服务 …… 第 1 章 1.5 节

购买《C#编程全能词典》的用户可享受四位一体服务（400 电话、QQ 服务、网站论坛支持、TQ）。

《C#编程全能词典》提供三种升级渠道（光盘升级、下载升级包本地升级、编程词典网在线升级）。升级内容包括软件本身升级及内容的少量升级。



更多资源请访问稀酷客(www.ckook.com)

目　录

第 1 部分　快速使用指南

第 1 章　为什么说《C#编程全能词典》是编程开发人员的必备工具　1

1.1　海量编程资源……2
1.1.1　超容量技术资源库……2
1.1.2　全方位视频资源库……2
1.1.3　丰富的实例资源库……3
1.1.4　多行业项目资源库……4
1.1.5　实际应用方案资源库……4
1.1.6　精美的界面素材库……5
1.1.7　详尽的源码资源库……6
1.1.8　多功能的工具库……6
1.2　《C#编程全能词典》适合各类人员使用……6
1.3　高效查询　智能检索……7
1.4　多种词典　协助编程……8
1.5　享有多种方式的升级与售后服务……9
1.5.1　享受全方位服务……10
1.5.2　提供多种方式的升级……10

第 2 章　怎样快速使用《C#编程全能词典》　12

2.1　启动《C#编程全能词典》……12
2.1.1　如何启动主程序……13

更多资源请访问稀酷客(www.ckook.com)

2.1.2 如何启动浮动窗口……13
2.2 初识程序界面……13
2.2.1 认识主导航区……14
2.2.2 了解功能导航区……15
2.2.3 了解内容导航区……16
2.2.4 了解内容显示设置区……18
2.3 特色功能快速使用……22
2.3.1 分类管理个人资源……22
2.3.2 编程资源查询……28
2.3.3 浮动窗口查询……31
2.3.4 收藏重要文档……32
2.3.5 记录学习心得……33
2.3.6 为文档设置书签……34
2.3.7 切换内容显示区……35

第 3 章 如何在“开发”使用模式中学习和使用编程资源 36

3.1 如何分类管理个人编程资源……37
3.2 如何学习浏览编程技术资源……38
3.3 如何学习浏览应用实例资源……43
3.4 如何学习浏览开发项目资源……48
3.5 如何学习浏览应用方案资源……51
3.6 观看入门及开发视频资源……54
3.6.1 程序入门学习录像……55
3.6.2 项目开发视频录像……56
3.7 浏览及使用界面素材资源……57
3.7.1 界面浏览……58
3.7.2 界面中心内容导读……60
3.7.3 通过目录导航浏览图片……62
3.8 根据实际需要使用“开发”模式……63
3.8.1 查询编程资源……64
3.8.2 学习编程技术……65
3.8.3 管理“我的代码”……66

更多资源请访问稀酷客(www.ckook.com)

第 4 章 使用“入门”模式进行阶段式学习+强化训练 68

4.1 分阶段学习编程技术 …… 68
4.1.1 从零开始 …… 69
4.1.2 进阶提高 …… 70
4.1.3 中级开发 …… 72
4.1.4 高级开发 …… 72
4.2 通过训练提高开发技能 …… 73
4.2.1 入门训练 …… 73
4.2.2 进阶训练 …… 75
4.2.3 中级训练 …… 75
4.2.4 高级训练 …… 76
4.3 不同水平的编程人员如何使用“入门”模式 …… 77
4.3.1 零基础编程人员学习方法 …… 77
4.3.2 中级开发人员使用方法 …… 77
4.3.3 高级开发人员使用方法 …… 78

第 5 章 如何使用“应用”模式实现高效开发 79

5.1 利用目录快速学习和检索编程技术 …… 80
5.2 高效查询与智能检索编程技术 …… 83
5.2.1 通过多种方式查询编程技术 …… 83
5.2.2 通过单个字母（关键字）智能检索编程技术 …… 88
5.3 有效利用源码实现选择性下载 …… 90
5.4 计算机（专业）英语速查 …… 91
5.4.1 搜索编程专用英语解释 …… 92
5.4.2 如何隐藏英语词典目录、添加标签 …… 93
5.5 使用多种辅助词典协助编程 …… 94
5.5.1 SQL 语句词典 …… 94
5.5.2 SQL 数据库技术词典 …… 95
5.5.3 UML 词典 …… 96
5.5.4 软件工程词典 …… 97
5.5.5 程序测试词典 …… 98

更多资源请访问稀酷客(www.ckook.com)

5.5.6 编码规范词典……99
5.5.7 术语词典……100
5.6 借助各种实用工具快速提高开发效率……101
5.7 有效通过小工具解决实际问题……102
5.8 不同人群如何使用“应用”模式进行快速、高效的开发……103
5.8.1 高效查询与检索海量编程资源……103
5.8.2 使用英语词典及其他辅助词典协助编程……104
5.8.3 借助各种实用工具实现高效开发……107

第 2 部分 速查手册

第 6 章 函数速查 108

A……108
Abort 方法……108
Abs 方法……108
AcceptButton 属性……109
Activate 事件……109
Add 方法……109
AddDays 方法……111
AddString 方法……111
AllowEdit 属性……112
AppendText 方法……112
Appllication 类……112
ArrayList 类……113
as 操作符……115
ASCII 码……115
ASCIIEncoding 类……116
Attributes 属性……117
AVG 聚合函数……118
B……119
BackColor 属性……119
BackgroudColor 属性……119

更多资源请访问稀酷客(www.ckook.com)

BackgroudImage 属性 119
BackgroundWorker 组件 120
base 关键字 121
BeginConnect 方法 122
BeginEdit 方法 122
BinaryReader 类 123
BinaryWriter 类 124
BindingNavigator 控件 125
Bitmap 类 126
Border 属性 127
BorderStyle 属性 128
break 语句 128
BringToFront 方法 128
Button 控件 129
变量 130
C 133
CalendarFont 属性 133
CalendarForeColor 属性 133
CancelButton 属性 133
CancelEdit 属性 134
CanFocus 属性 134
CanPaste 方法 134
Capture 属性 135
case 语句 135
CAST 函数 136
Cell 对象 136
CellClick 事件 136
CellEnter 事件 137
CellMouseClick 事件 137
CellLeave 事件 137
Cells 属性 137
CheckBox 控件 138

更多资源请访问稀酷客(www.ckook.com)

Checked 属性……138
CheckedChanged 事件……139
Children 属性……139
Clear 方法……140
Click 事件……140
Clone 方法……140
Close 方法……141
Color 属性……141
ColorDialog 控件……142
ColumnCount 属性……142
Columns 属性……143
ComboBox 控件……143
Command 对象……144
CommandText 属性……146
CommandType 属性……146
Compare 方法……147
CompareTo 方法……147
Computer 类……148
ComputerInfo 类……149
Connection 对象……151
ConnectionState 枚举……151
ContextMenuStrip 控件……152
continue 语句……152
Control 类……152
Convert 类……155
Console 类……156
Copy 方法……158
CopyFile 方法……158
COUNT 函数……159
Count 属性……159
Create 方法……160
CreateDirectory 方法……160

更多资源请访问稀酷客(www.ckook.com)

CreateSubKey 方法 …… 161
CreateText 方法 …… 161
CryptoStream 类 …… 162
CrystalReportViewer 控件 …… 163
CurrentCell 属性 …… 165
CurrentRow 属性 …… 165
Cursor 类 …… 166
Cut 方法 …… 166
常量 …… 167
D …… 168
DataAdapter 类 …… 168
DataGridView 控件 …… 169
DataMember 属性 …… 171
DataReader 类 …… 172
DataSet 类 …… 173
DataSource 属性 …… 174
DATEADD 函数 …… 174
DateDiff 方法 …… 175
DateTime 结构 …… 176
DateTimePicker 控件 …… 177
DAY 函数 …… 178
DayOfWeek 属性 …… 179
Delete 方法 …… 179
Directory 类 …… 180
DirectoryEntry 类 …… 181
DirectoryInfo 类 …… 183
DirectoryName 属性 …… 183
Dispose 方法 …… 184
Dns 类 …… 184
do…while 语句 …… 185
Dock 属性 …… 185
Draw 方法 …… 186

更多资源请访问稀酷客(www.ckook.com)

DrawArc 方法……186
DrawEllipse 方法……187
DrawImage 方法……188
DrawLine 方法……189
DrawPath 方法……190
DrawPie 方法……191
DrawPolygon 方法……192
DrawRectangle 方法……193
DrawString 方法……194
DriveInfo 类……195
DropDownStyle 属性……196
迭代器……197
E……198
Enabled 属性……198
Encoding 属性……198
Enter 事件……199
Environment 类……199
Equals 方法……200
ErrorProvider 控件……202
EventLog 组件……203
ExecuteNonQuery 方法……204
ExecuteReader 方法……205
Exists 方法……206
Exists 属性……207
ExpandAll 方法……207
Extension 属性……208
F……209
File 类……209
FileInfo 类……210
FileName 属性……211
FileStream 类……212
Fill 方法……213

更多资源请访问稀酷客(www.ckook.com)

FillEllipse 方法……214
FillPath 方法……215
FillPie 方法……216
FillPolygon 方法……217
FillRectangle 方法……218
FillRegion 方法……219
Filter 属性……220
Find 方法……220
First 函数……221
FirstDayOfWeek 属性……222
FlatStyle 属性……222
FlowDirection 属性……223
FlowLayoutPanel 控件……223
Flush 方法……224
Focus 方法……225
FolderBrowserDialog 控件……225
Font 属性……226
FontDialog 控件……227
for 语句……228
foreach 语句……228
ForeColor 属性……229
Form 窗体……229
Format 方法……231
FormClosed 事件……233
FormClosing 事件……233
FromArgb 方法……233
泛型……235
分部类……236
G……239
GDI+……239
GetBytes 方法……239
GetDirectories 方法……239

更多资源请访问稀酷客(www.ckook.com)

GetFiles 方法……240
GetFileSystemEntries 方法……240
GetFileSystemInfos 方法……240
GetHostAddresses 方法……241
GetHostByAddress 方法……241
GetHostByName 方法……242
GetHostName 方法……242
GetPixel 方法……242
GetString 方法……243
GetValueNames 方法……243
goto 语句……243
Graphics 类……244
GroupBox 控件……248
GetValue 方法……249
H……250
Handle 属性……250
Hashtable 类……250
HasRows 属性……253
Height 属性……253
HelpProvider 组件……254
Hide 方法……256
HScrollBar 控件……256
I……259
IsUpper 方法……259
ImageFormat 类……259
IsLeapYear 方法……260
Image 属性……260
ImageList 组件……260
IndexOf 方法……261
InitialDirectory 属性……262
Insert 方法……262
Interval 属性……263

更多资源请访问稀酷客(www.ckook.com)

IPEndPoint 类 263
IPHostEntry 类 265
IsMatch 方法 265
IsMdiContainer 属性 266
Items 属性 266
J 267
Join（String 类）方法 267
Join（Thread 类）方法 267
K 268
KeyChar 属性 268
KeyDown 事件 268
KeyPress 事件 269
KeyUp 事件 269
Kill 方法 269
L 271
Label 控件 271
LabelEdit 属性 272
Last 函数 273
LastIndexOf 方法 273
LayoutMdi 方法 274
Leave 事件 274
Lines 属性 274
LinkLabel 控件 275
ListBox 控件 276
ListView 控件 278
Load 方法 280
LoadFile 方法 280
Location 属性 281
M 282
MachineName 属性 282
Macthes 方法 283
MailMessage 类 284

更多资源请访问稀酷客(www.ckook.com)

ManagementClass 类……285
ManagementObject 类……286
MaskedTextBox 控件……287
MAX 函数……288
MaxDate 属性……289
Maximum 属性……289
MDI 窗体……290
MdiChildren 属性……293
MdiParent 属性……294
Microsoft.Win32 命名空间……294
MIN 函数……295
MinDate 属性……295
Minimum 属性……296
MonthCalendar 控件……296
MouseClick 事件……297
Move 方法……297
MoveTo 方法……298
MVC 开发模式……299
N……302
Navigated 事件……302
Navigate 方法……302
NetworkStream 类……303
new 运算符……304
newid 函数……304
NewRow 方法……305
Next 方法……305
NotifyIcon 控件……305
NumericUpDown 控件……307
Net send 命令……307
Net use 命令……309
O……311
object 类型……311

更多资源请访问稀酷客(www.ckook.com)

Opacity 属性 …… 312
Open 方法 …… 313
OpenFile 方法 …… 314
OpenFileDialog 控件 …… 314
OpenSubKey 方法 …… 316
OpenText 方法 …… 316
Owner 属性 …… 317
P …… 318
Paint 事件 …… 318
Panel 控件 …… 318
PasswordChar 属性 …… 319
Path 属性 …… 320
PictrueBox 控件 …… 321
POP3 协议 …… 324
Position 属性 …… 326
Print 方法 …… 327
PrintDialog 组件 …… 328
PrintDocument 组件 …… 330
PrintPreviewControl 组件 …… 332
PrintPreviewDialog 组件 …… 333
Process 类 …… 334
ProgressBar 控件 …… 337
Ping 类 …… 338
PrinterSettings 类 …… 341
Q …… 343
Queue 类 …… 343
R …… 344
RadioButton 控件 …… 344
Random 类 …… 345
Read 方法 …… 346
ReadBytes 方法 …… 347
ReadLine 方法 …… 348

更多资源请访问稀酷客(www.ckook.com)

ReadOnly 属性……348
ReadToEnd 方法……349
ReadXml 函数……350
Receive 方法……350
Refresh 函数……351
Regex 类……351
Registry 类……352
RegistryKey 类……353
Remove 方法……355
RemoveAt 方法……356
Replace 方法……357
Return 语句……359
RichTextBox 控件……360
Right 属性……362
Round 方法……363
RowCount 方法……363
Rows 属性……363
S……366
Save 方法……366
SaveFile 方法……366
SaveFileDialog 组件……367
ScrollBars 属性……368
Select 方法……369
SelectAll 方法……371
Selected 属性……372
SelectedCells 属性……372
SelectedColumns 属性……373
SelectedIndex 属性……373
SelectedItem 属性……374
SelectedItems 属性……374
SelectedPath 属性……375
SelectedRows 属性……375

更多资源请访问稀酷客(www.ckook.com)

SelectedText 属性 ····· 377
SelectionColor 属性 ····· 377
SelectionEnd 属性 ····· 378
SelectionFont 属性 ····· 378
Send 方法 ····· 379
SetDate 方法 ····· 381
SetError 方法 ····· 381
SetPixel 方法 ····· 382
SetToolTip 方法 ····· 382
SetValue 方法 ····· 383
Show 方法 ····· 384
ShowDialog 方法 ····· 385
ShowToday 属性 ····· 386
Sleep 方法 ····· 386
SMTP 协议 ····· 386
SmtpClient 类 ····· 389
Socket 类 ····· 390
SolidBrush 类 ····· 391
Sort 方法 ····· 392
Split 方法 ····· 393
SQL 注入式攻击 ····· 394
SqlCommand 类 ····· 395
SqlConnection 类 ····· 396
SqlDataAdapter 类 ····· 398
SqlDataReader 类 ····· 399
Stack 类 ····· 401
Start 方法 ····· 402
StartPosition 属性 ····· 403
Step 属性 ····· 403
Stop 方法 ····· 404
Stream 类 ····· 404
StreamReader 类 ····· 405

更多资源请访问稀酷客(www.ckook.com)

StreamWriter 类……406
String 类……407
StringBuilder 类……409
Substring 方法……410
SUM 函数……410
switch case 语句……411
T……412
TabControl 控件……412
TabIndex 属性……413
Tables 属性……414
TabPages 属性……414
Tag 属性……415
TcpClient 类……415
TcpListener 类……416
Text 属性……417
TextBox 控件……417
TextChanged 事件……418
TextReader 类……419
TextWriter 类……419
Thread 类……420
ThreadState 属性……422
throw 语句……423
Tick 事件……423
Timer 组件……423
TimeSpan 对象……424
ToInt32 方法……426
ToLower 方法……426
ToolStrip 控件……427
ToolTip 控件……428
ToolTipIcon 属性……429
ToolTipText 属性……429
ToolTipTitle 属性……430

更多资源请访问稀酷客(www.ckook.com)

Top 属性 …… 430
TopMost 属性 …… 431
ToShortDateString 方法 …… 431
ToShortTimeString 方法 …… 431
ToString 方法 …… 432
ToUpper 方法 …… 432
TrackBar 控件 …… 433
TreeNode 类 …… 435
TreeView 控件 …… 436
try…catch…finally 语句 …… 437
typeof 运算符 …… 438
U …… 439
UdpClient 类 …… 439
UpButton 方法 …… 440
Update 方法 …… 440
UPDATE 语句 …… 441
UpdateCommand 属性 …… 442
Url 属性 …… 442
using 关键字 …… 443
V …… 444
Value 属性 …… 444
View 属性 …… 444
Visible 属性 …… 445
VscrollBar 控件 …… 445
W …… 447
WebBrowser 控件 …… 447
WebClient 类 …… 451
While 语句 …… 453
Width 属性 …… 454
WindowState 属性 …… 454
WndProc 方法 …… 455
Write 方法 …… 455

更多资源请访问稀酷客(www.ckook.com)

WriteLine 方法 ……460
WriteXml 方法 ……462
X ……463
XML ……463
XmlReader 类 ……465
XmlWriter 类 ……467
Y ……469
YEAR 函数 ……469
Z ……470
Zoom 属性 ……470

更多资源请访问稀酷客(www.ckook.com)

第 1 章

为什么说《C#编程全能词典》是编程开发人员的必备工具

《C#编程全能词典》软件产品是由吉林省明日科技有限公司组织数十名C#资深技术人员，历经数年积累开发的编程辅助软件。

《C#编程全能词典》软件以其特有的学习模式、全面的资源（技术资源库、视频库、实例库、项目库、界面库、工具库等）和高效的查询与检索功能深受编程爱好者的喜欢。无论你是初学编程的爱好者、大中专院校的师生、初、中级程序设计人员，还是具有编程经验的开发人员和从事软件研发的专家，使用《C#编程全能词典》软件都可以让你轻松上手、快速编程、高效开发。

《C#编程全能词典》软件的特点如图 1.1 所示。

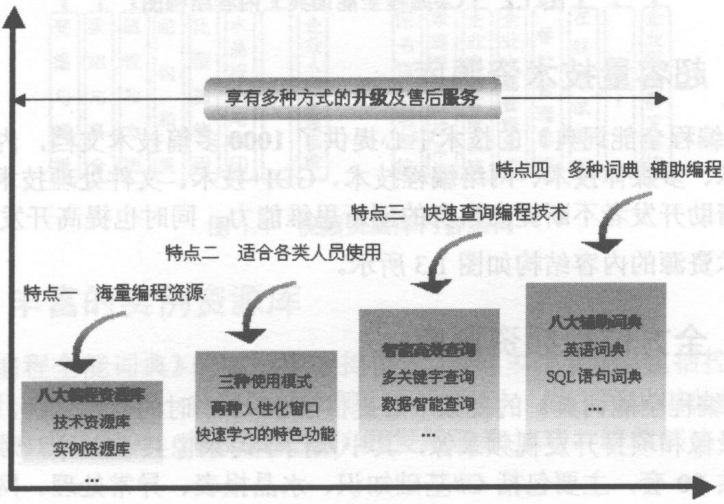


图 1.1　《C#编程全能词典》软件的特点

更多资源请访问稀酷客(www.ckook.com)

1.1 海量编程资源

《C#编程全能词典》软件提供了 8 大编程资源库，包括技术资源库、实例资源库、视频讲解库、方案资源库、项目资源库、源码资源库、界面素材资源库和工具资源库。通过 8 大资源库编程人员既可以分类学习和使用编程技术，也可以通过软件提供的高效查询和智能检索查询编程技术。

《C#编程全能词典》软件的内容结构图如图 1.2 所示。

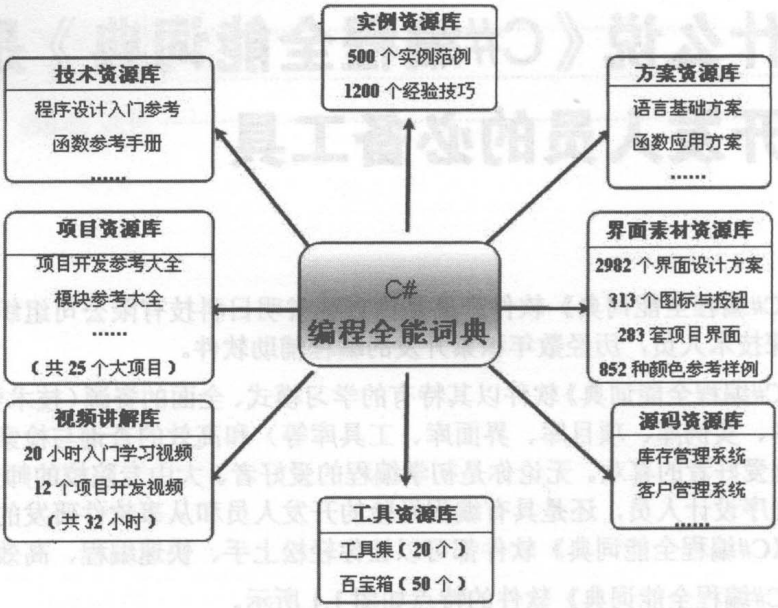


图 1.2 “C#编程全能词典”内容结构图

1.1.1 超容量技术资源库

《C#编程全能词典》的技术中心提供了 **1000** 多篇技术文档，内容包括数据库技术、多媒体技术、网络编程技术、GDI+技术、文件处理技术、打印技术等。帮助开发者不断提高自身的创新思维能力，同时也提高开发效率。

技术资源的内容结构如图 1.3 所示。

1.1.2 全方位视频资源库

《C#编程全能词典》的视频中心提供了 **30** 个小时的视频讲解，分程序入门学习录像和项目开发视频录像，其中入门学习录像共 **112** 个，项目开发视频录像共 **10** 套，主要包括 C#基础知识、水晶报表、异常处理、网络编程技术及图书馆管理系统、企业短信群发系统、企业人事管理系统等视频讲解，

更多资源请访问稀酷客(www.ckook.com)

让编程人员快速掌握编程技巧、全方位掌握项目开发流程。

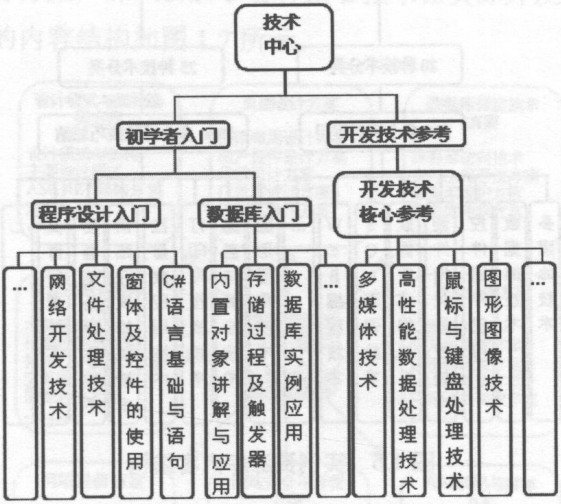


图 1.3 技术资源库内容结构

其内容结构如图 1.4 所示。

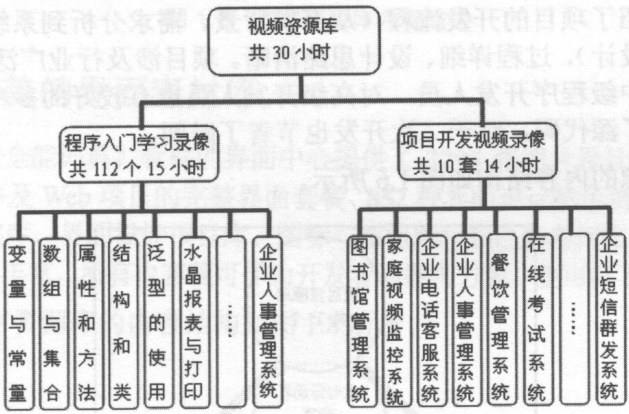


图 1.4 视频资源库内容结构

1.1.3 丰富的实例资源库

《C#编程全能词典》的实例中心提供了 **1500** 多个实例，包括控件应用、组件应用、图形技术、多媒体技术等 **45** 个类别，用户可以以此借鉴其他开发人员的成功案例，并积累大量的开发经验和技巧，培养解决实际问题的能力。

实例资源的内容结构如图 1.5 所示。

更多资源请访问稀酷客(www.ckook.com)

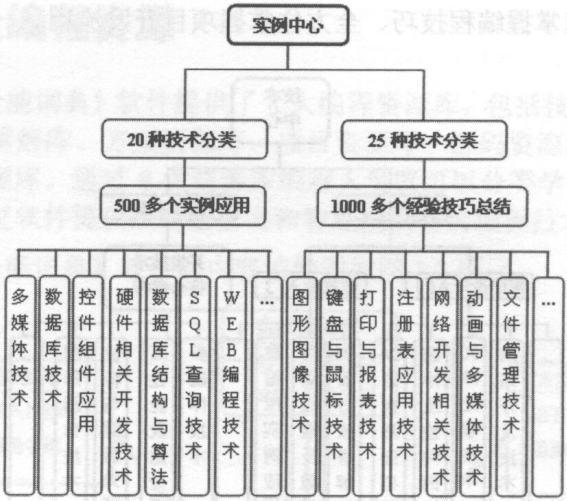


图 1.5　实例资源库内容结构

1.1.4　多行业项目资源库

《C#编程全能词典》软件的项目中心提供了 **25** 个大项目，包括项目开发参考大全和模块参考大全，从软件工程的角度出发，按项目的开发顺序，系统全面地介绍了项目的开发流程（从开发背景、需求分析到系统功能分析、前后台页面设计），过程详细、设计思路清晰。项目涉及行业广泛，实用性强，适合于初、中级程序开发人员，对高级开发人员也有较好的参考价值。由于案例中提供了源代码，对于二次开发也节省了时间。

项目资源的内容结构如图 1.6 所示。

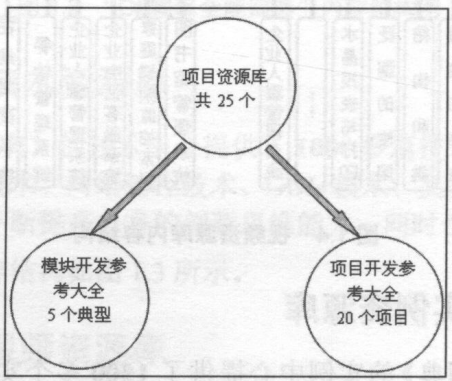


图 1.6　项目资源库内容结构

1.1.5　实际应用方案资源库

《C#编程全能词典》软件的方案中心提供了 **145** 个实际应用的方案，主要

更多资源请访问稀酷客(www.ckook.com)

包括页面设计、数据库开发技术、水晶报表打印、决策分析、网络通信、网站安全与管理等方案，深入讲解了各种核心技术在实际开发中的应用。

方案资源的内容结构如图 1.7 所示。

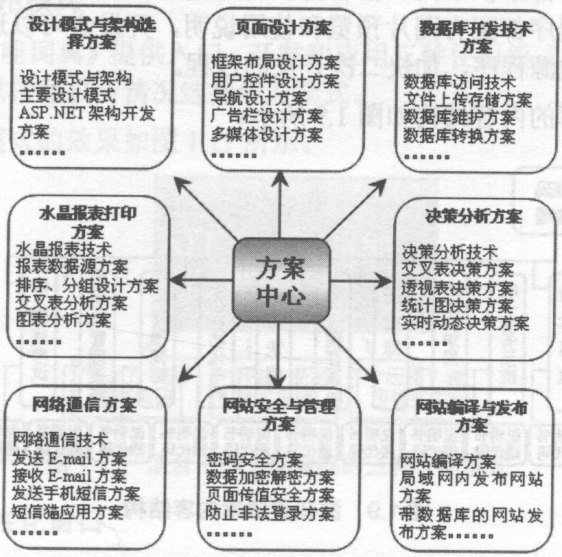


图 1.7　方案资源库内容结构

1.1.6　精美的界面素材库

《C#编程全能词典》软件的界面中心提供了 **3295** 张图片设计方案、**283** 套界面应用程序及 Web 项目的完整界面套餐、**852** 种界面设计颜色例库，主要包括界面设计方案库、界面设计项目库、图标与按钮库和颜色参考例库，界面素材覆盖面广、内容丰富，所有内容既可作为开发者的参考方案，也可以下载直接使用。

界面素材资源库的内容结构如图 1.8 所示。

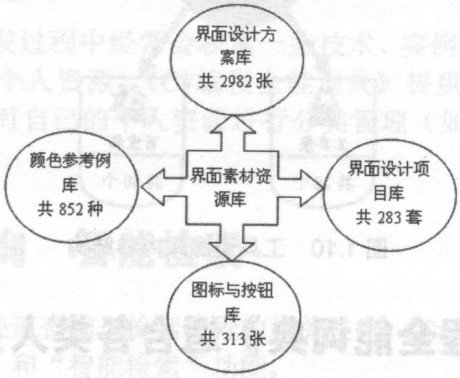


图 1.8　界面素材资源库内容结构

更多资源请访问稀酷客(www.ckook.com)

1.1.7 详尽的源码资源库

《C#编程全能词典》提供了 **10** 套完整程序的源码，主要包括博客系统、电子商务系统、都市供求网、留言簿、企业进销存管理系统、企业快信等程序源码，每套程序都配有图片预览和使用说明，开发者可以通过阅读说明书有选择性的下载源程序，加快二次开发的过程。

源码资源库的内容结构如图 1.9 所示。

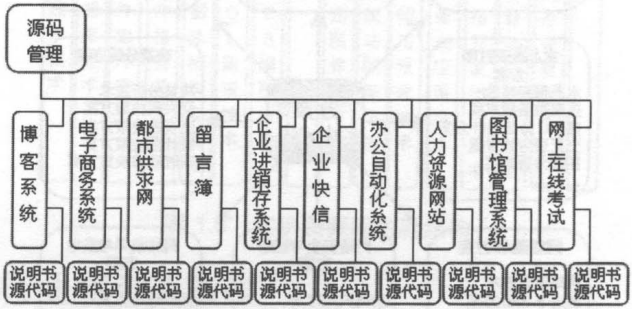


图 1.9 源码资源库内容结构

1.1.8 多功能的工具库

《C#编程全能词典》软件提供了 **70** 个实用小工具，其中工具集 **20** 个，百宝箱 **50** 个，包括 SQL 数据库提取器、备份数据源、获取局域网计算机名称、监控上网过程的程序等实用工具，开发者可通过这些工具来解决编程中遇到的某些实际性的问题。

工具资源库的内容结构如图 1.10 所示。

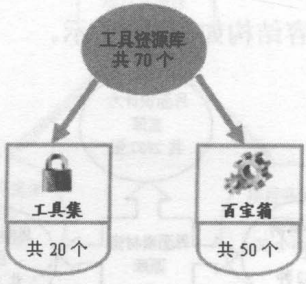


图 1.10 工具资源库内容结构

1.2 《C#编程全能词典》适合各类人员使用

三种适合不同人群的使用模式；两种人性化的窗口设计；辅助编程人员

更多资源请访问稀酷客(www.ckook.com)

快速使用《C#编程全能词典》软件的特色功能（书签、收藏、心得、显示源程序等），这些都可以让各类人员在使用《C#编程全能词典》软件时轻松上手、快速编程。

1. 三种使用模式

《C#编程全能词典》提供入门、开发和应用三种使用模式，适用于不同的人群，用户可以根据自身情况选择学习模式。

三种使用模式的效果如图 1.11 所示。

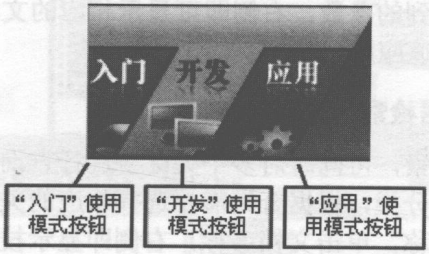


图 1.11　适合不同人群需要的三种使用模式

2. 两种人性化窗口

《C#编程全能词典》软件提供了两种窗口：主窗口和浮动窗口。

其中浮动窗口开机时会自动运行，隐藏于屏幕上方，不妨碍正常工作。适用于编程人员工作时使用。

3. 提供快速学习的特色功能

《C#编程全能词典》软件为方便编程人员快速使用、学习和查询编程技术，提供了书签、收藏、心得、切换等快捷按钮；同时“显示源程序”按钮可以复制源程序代码，加快二次开发过程。

4. 管理个人资源

编程人员在开发过程中经常会积累一些技术、实例、图片等编程资源，为有效管理这部分个人资源，《C#编程全能词典》提供了“我的代码”版块。编程人员可以对自己的个人资源进行分类管理（如查看、删除、修改等）。

1.3　高效查询　智能检索

为方便开发者快速查询与检索编程的相关技术，《C#编程全能词典》软件提供了“高效查询”和“智能检索”功能。

更多资源请访问稀酷客(www.ckook.com)

1. 多关键字查询

多关键字查询可以同时对多个关键字进行查询（如 where、go、list 等）。查询时还可以设置查询方式（“与”或“或”）和查询范围，查询结果将按查询分类显示，并显示各中心中查询结果的记录条数，查询关键字也将以不同颜色的描红方式显示。

2. 数据智能查询

数据智能查询可以将所查到的数据按标题和内容进行分类，并且对于每一种分类显示所查询到的条数，右侧即可显示相应的文档，关键字以描红方式显示，方便用户快速应用。

3. 自动批量数据检索

自动批量数据检索，可同时对多个关键字进行查询，检索到的文档按照所设置的关键字进行分类，根据查询需要选择相应的关键字，下方目录列出了该关键字的文档名称，单击文档名称，右侧即显示技术文档，其中关键字以描红方式显示。

4. 自动提取关键数据

自动提取关键数据能够快速、准确地将关键字相关的关键代码提炼出来，代码应用一目了然。用户可以根据关键数据学习和应用关键字的使用，关键字以描红方式显示。

5. 智能检索

智能检索的所有内容名称都是按照字母的先后顺序排列的，随着用户每输入一个关键字（字母），即可快速检索到包含其关键字的所有文档资料。在一定程度上大大提高了检索速度及检索的深度。

1.4 多种词典 协助编程

《C#编程全能词典》软件提供了 8 种辅助词典，协助开发人员快速编程。包括英语词典、SQL 语句词典、编码规范词典、UML 词典、SQL 数据库技术词典、软件工程词典、程序测试词典、术语词典。用户可随用随查，快速解决开发难题。

8 种辅助词典的界面效果图如图 1.12 所示。

1. 英语词典

“英语词典”收录了编程时使用频率最高的计算机英语及专业英语的汉语解释，可对用户搜索的单词进行模糊查询。

更多资源请访问稀酷客(www.ckook.com)

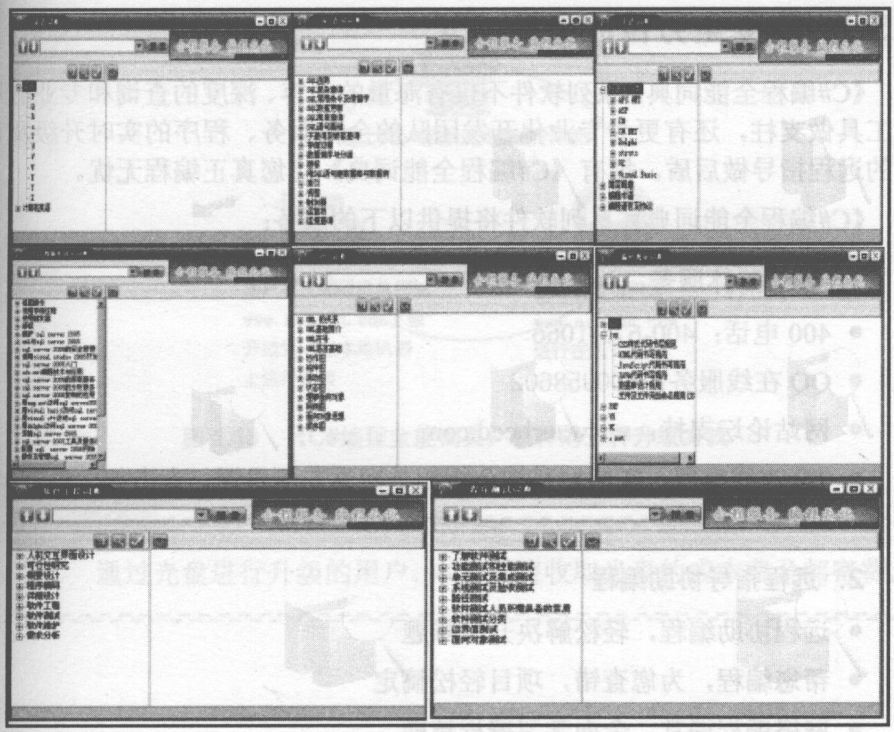


图 1.12　8 种辅助词典

2. 其他辅助词典

除英语词典外，其他 7 种辅助词典（包括 SQL 语句词典、SQL 数据库技术词黄、UML 词典、软件工程词典等）内容涉及 SQL 语句、数据库技术、编码规范等。通过这些辅助词典可以协助编程人员快速编程。

其中“SQL 语句词典”包括了所有与数据库相关的 SQL 语句。编程人员可通过“SQL 语句词典”查询与之相关的所有 SQL 语句（如 WHERE、SELECT……）。

“软件工程词典”对软件开发前期的分析、开发过程中的设计，以及后期的维护和测试都有详细的说明。开发人员通过查询对软件的开发会有一个清晰的认识，开发出好的、实用的软件。

1.5　享有多种方式的升级与售后服务

对购买《C#编程全能词典》系列软件的用户，明日科技有限公司将提供全方位、零距离的售后服务；同时提供《C#编程全能词典》系列软件的三种升级方式，升级内容包括软件本身的升级，以及《C#编程全能词典》软件内容的升级。

更多资源请访问稀酷客(www.ckook.com)

1.5.1 享受全方位服务

《C#编程全能词典》系列软件不仅有海量的内容、深度的查询和专业的开发工具做支柱，还有更为专业化开发团队的全程服务、程序的实时升级及专家的远程指导做后盾。拥有《C#编程全能词典》让您真正编程无忧。

《C#编程全能词典》系列软件将提供以下的服务：

1. 四位一体服务

- 400 电话：400-675-1066
- QQ 在线服务：200958602
- 网站论坛支持：www.mrbccd.com
- TQ 在线客服：在明日科技公司相关网站左侧悬浮，点击相关语言 TQ 即可

2. 远程指导协助编程

- 远程协助编程，轻松解决开发难题
- 帮您编程，为您查错，项目轻松搞定
- 网络编程课堂，全面学习编程技能



提示

《C#编程全能词典》软件提供四位一体服务，《C#编程全能词典》豪华版除提供四位一体服务以外还将提供远程指导，协助编程的服务。

1.5.2 提供多种方式的升级

购买《C#编程全能词典》软件以后，用户便可以通过明日科技公司提供的多种升级方式对《C#编程全能词典》软件进行升级。

购买《C#编程全能词典》软件的版本不同，提供的升级服务也有所差别。《C#编程全能词典》将提供一年 5 次的软件本身的升级，以及内容的少量升级；《C#编程全能词典》标准版将提供一年 5 次的免费升级，同时提供软件内容的升级（包括技术人员最新开发的技巧、项目、实例、方案、录像等）。

明日科技公司将提供以下三种升级方式，如图 1.13 所示。通过本公司提供的光盘进行升级；从编程全能词典网站 www.mrbccd.com 下载升级包，在本地机器上进行升级；登录编程全能词典网站 www.mrbccd.com 进行在线升级。用户可以根据情况选择任意一种升级方式进行升级。

更多资源请访问稀酷客(www.ckook.com)

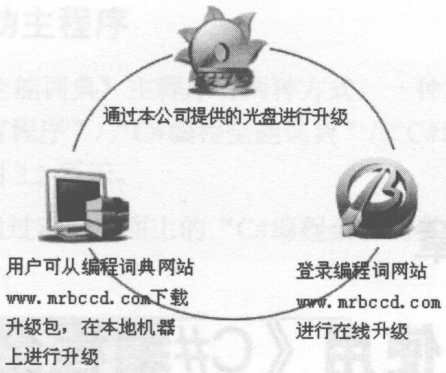


图 1.13 《C#编程全能词典》软件的三种升级方式

注意

通过光盘进行升级的用户，本公司将收取光盘的成本费及邮寄费。

更多资源请访问稀酷客(www.ckook.com)

第 2 章

怎样快速使用《C#编程全能词典》

《C#编程全能词典》软件产品是由吉林省明日科技有限公司组织数十名C#资深技术人员，历经数年积累开发的编程辅助软件。

该软件具有最全面的编程资源；最高效快捷的智能检索方式；最完善的服务支持、最超值的产品升级；三种使用模式，覆盖各类用户人群；个人资源管理器，有效管理个人编程资源；英语词典、辅助词典、工具集、百宝箱，使编程更高效，《C#编程全能词典》软件是程序开发人员的必备工具。

《C#编程全能词典》软件，您的私人编程专家！

快速使用《C#编程全能词典》的效果图如图 2.1 所示。

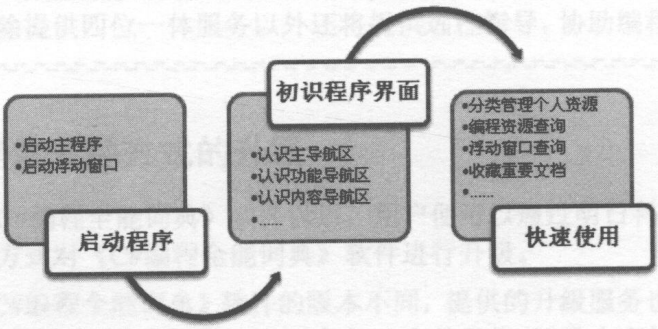


图 2.1　快速使用《C#编程全能词典》

2.1　启动《C#编程全能词典》

《C#编程全能词典》软件包含主程序和浮动窗口两部分，主程序可以通过菜单命令和快捷方式图标两种方式启动；浮动窗口可以通过开机自动启动、菜单命令和托盘程序三种方式启动。

更多资源请访问稀酷客(www.ckook.com)

2.1.1 如何启动主程序

启动《C#编程全能词典》主程序有两种方式，一种方式是通过选择“开始”菜单中的“所有程序”/“C#编程全能词典”/“C#编程全能词典”菜单命令进行启动，如图 2.2 所示。

另一种方式是通过双击桌面上的“C#编程全能词典”快捷方式图标进行启动，如图 2.3 所示。

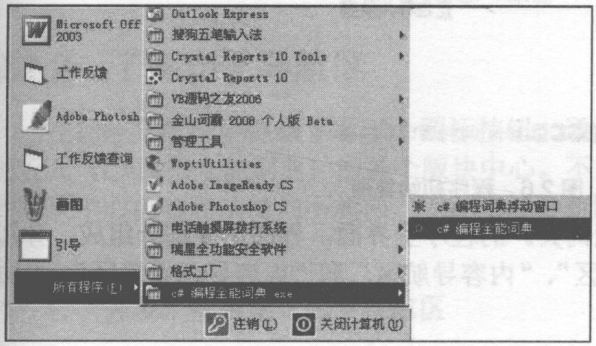


图 2.2 通过菜单命令启动主程序

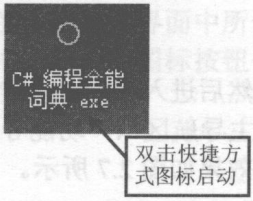


图 2.3 通过快捷方式图标启动主程序

2.1.2 如何启动浮动窗口

《C#编程全能词典》的浮动窗口是编程词典的辅助程序，该程序主要用于方便、快速地查询。浮动窗口有三种启动方式，一种是在安装《C#编程全能词典》软件时设置为开机自动启动；一种是通过选择“开始”菜单中的“所有程序”/“C#编程全能词典”/“C#编程词典浮动窗口”菜单命令进行启动，如图 2.4 所示。

第三种是通过单击系统任务栏通知区域中的托盘程序图标启动，如图 2.5 所示。

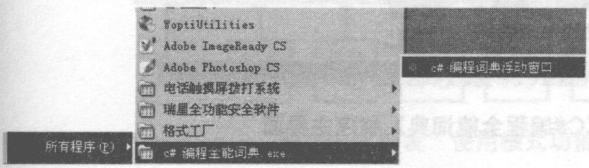


图 2.4 通过菜单命令启动浮动窗口

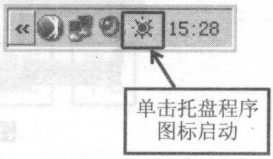


图 2.5 通过托盘程序图标启动浮动窗口

2.2 初识程序界面

启动《C#编程全能词典》软件以后，首先显示的是程序初始界面，如图

更多资源请访问稀酷客(www.ckook.com)

2.6 所示。



图 2.6　程序初始界面

然后进入《C#编程全能词典》的程序主界面，界面由 4 部分组成，分别为“主导航区”、“功能导航区”、“内容导航区”和“内容显示设置区”。界面运行效果如图 2.7 所示。



图 2.7 《C#编程全能词典》程序主界面

2.2.1　认识主导航区

《C#编程全能词典》针对不同的用户人群设计了三种使用模式，分别为：“入门”、“开发”和“应用”。通过单击主导航中的按钮即可切换到相应的使用模式界面。主导航区如图 2.8 所示。

更多资源请访问稀酷客(www.ckook.com)

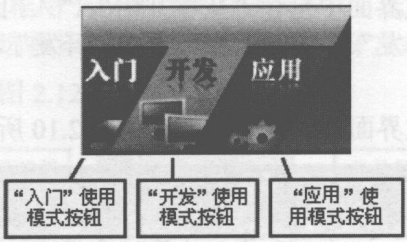


图 2.8 主导航区

2.2.2 了解功能导航区

在功能导航区中提供了若干个图标按钮，通过单击这些图标按钮可以快速进入《C#编程全能词典》的各个版块中心。不同的使用模式界面中所包含的版块中心内容和功能均不相同，因此功能导航区中所显示的图标按钮也不相同。

1.“开发”使用模式功能导航区

“开发”使用模式针对初、中级程序开发人员进行设计，其中“我的代码”版块用于分类管理用户的个人编程资源，另外还包含了编程所需技术、实例、项目等各类编程资源，是目前国内最全面、最系统的编程资源库，既适合深入学习编程技术，又适合在编程过程中查询使用。

“开发”使用模式界面中包含“我的代码”、“技术中心”、“视频中心”、“实例中心”、“项目中心”、“方案中心”、“界面中心”和“图书中心”8 个版块。其中“我的代码”版块用于管理用户的个人编程资源，“技术中心”、“界面中心”为编程资源的分类版块。

“开发”使用模式界面的功能导航区，如图 2.9 所示。

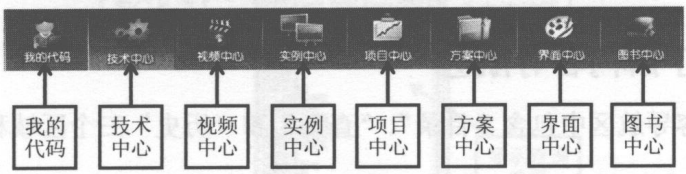


图 2.9 “开发”使用模式功能导航区

2.“入门”使用模式功能导航区

“入门”使用模式主要针对没有基础、初学编程的用户群设计，将学习C#编程所需的各项技术，按照由浅入深的原则划分为 4 个阶段，并且不同的阶段都提供了相应的实践练习题及答案，用于巩固和检验用户掌握编程技术的程度。

更多资源请访问稀酷客(www.ckook.com)

“入门”使用模式界面中包含“从零开始”、“入门训练”、“进阶提高”、“进阶训练”、“中级开发”、“中级训练”、“高级开发”和“高级训练”8个版块的内容。

“入门”使用模式界面的功能导航区，如图2.10所示。

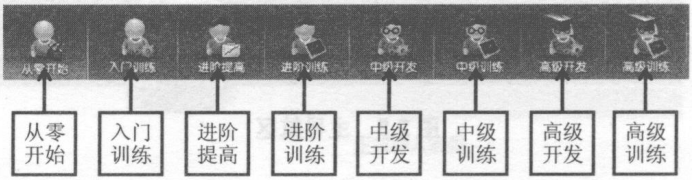


图2.10 “入门”使用模式功能导航区

3.“应用”使用模式功能导航区

“应用”使用模式主要针对中高级程序开发人员设计。在该使用模式中将“开发”使用模式中的“技术中心”、“实例中心”、“项目中心”及“方案中心”的内容合并到“目录”版块中，并提供了更加高效智能的查询及检索方式，便于编程人员在编程过程中查询使用。另外还提供了“英语辞典”、“辅助词典”及数十种实用工具，帮助编程人员高效开发。

“应用”使用模式界面中包含“目录”、“查询”、“检索”、“源码管理”、“英语辞典”、“辅助词典”、“工具集管理”和“百宝箱”8个版块。

“应用”使用模式界面的功能导航区，如图2.11所示。

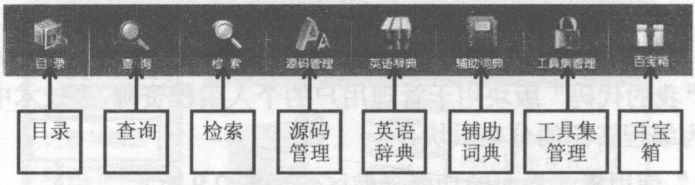


图2.11 “应用”使用模式功能导航区

2.2.3 了解内容导航区

在内容导航区中包含“目录”、“查询”和“历史”三个区域和“自动执行”按钮。

1. 目录显示区

在功能导航区中单击某个图标按钮以后，在内容导航区的目录显示区域中将显示该版块中内容的目录。

在区域上方显示当前所选择的版块。窗格下方的“单击执行”复选框用于设置打开文档的方式。选择该复选框，则通过单击目录中的文档名称打开文档；如果没有选择该复选框，则需双击文档名称将文档打开。

更多资源请访问稀酷客(www.ckook.com)

“手动展开”复选框用于设置目录的显示方式，选择该复选框，则目录呈折叠方式显示；没有选择该复选框，则目录呈展开方式显示。

内容目录区，如图 2.12 所示。

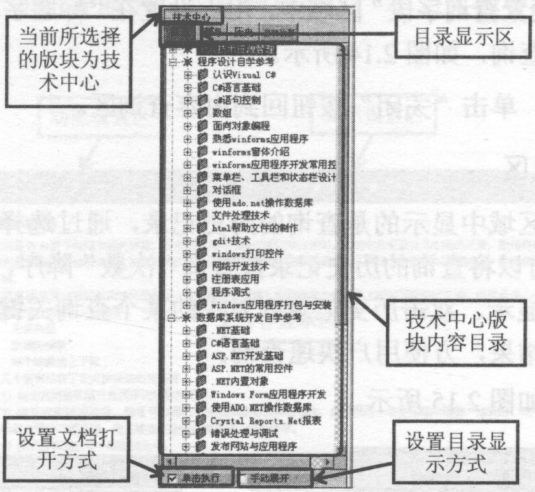


图 2.12　内容目录区

2. 内容查询区

单击“查询”按钮切换到内容查询区，通过该区域可以在当前中心及所有中心中进行查询。其中的文本框用于输入查询关键字，输入查询关键字以后，单击“搜索”按钮，或者直接按 Enter 键，在下面的列表框中将显示查询的结果，如图 2.13 所示。

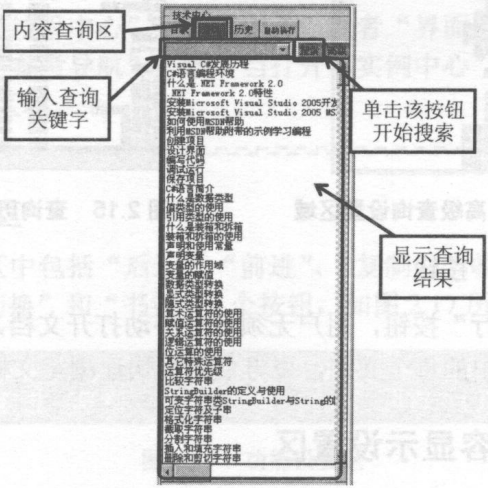


图 2.13　内容查询区

更多资源请访问稀酷客(www.ckook.com)

在内容查询区中，单击“高级”按钮，将显示查询设置选项。在“设置范围选择”区域中可以选择查询的范围，当前中心或者是所有中心；在“查询结果包含关键字设置”区域中，用于设置查询结果中是否显示查询结果所在的位置；在“设置查询字段”区域中，用于设置在“标题字段”，还是在“所有字段”中进行查询，如图 2.14 所示。

设置完成后，单击“关闭”按钮回到内容查询区。

3. 查询历史区

在查询历史区域中显示的是查询的历史记录，通过选择“次数”和“日期”单选按钮，可以将查询的历史记录按查询“次数”降序，或者按查询“日期”的先后进行显示。双击历史记录列表中的某个查询关键字，将快速显示该关键字的查询结果，方便用户快速查询。

查询历史区如图 2.15 所示。

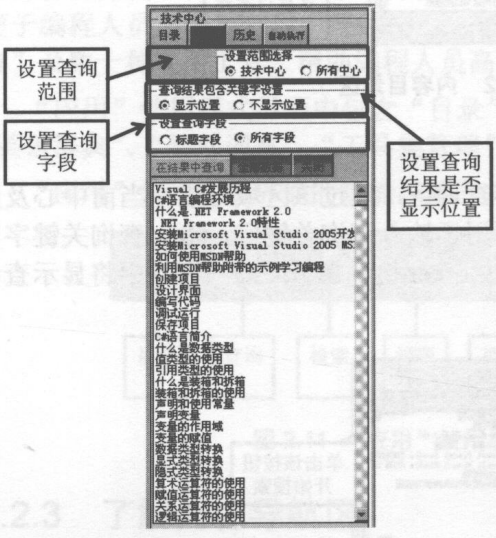


图 2.14　高级查询设置区域

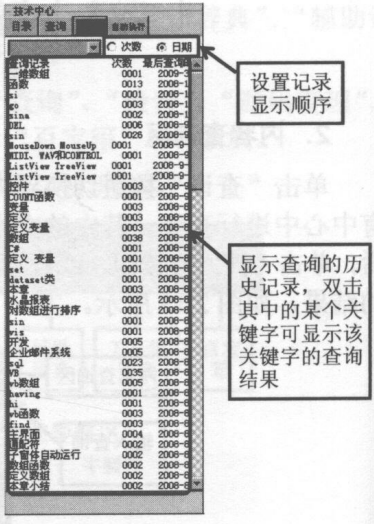


图 2.15　查询历史区

4.“自动执行”按钮

单击“自动执行”按钮，用户无须通过手动打开文档，程序将自动逐一打开，并显示目录中的每个文档，以方便用户快速浏览文档内容。

2.2.4　了解内容显示设置区

展开目录中的节点，选择最终节点的文档名称以后单击或双击，在内容

更多资源请访问稀酷客(www.ckook.com)

显示设置区中将显示文档的具体内容。在该区域中除了可以显示文档的具体内容以外，还可以进行相应的功能及显示设置。

内容显示设置区域由 6 部分组成，分别为“功能设置区”、“显示设置区”、“内容显示区”、“参考导航区”、“实例导引区”和“文档与源代码切换区”。如图 2.16 所示。

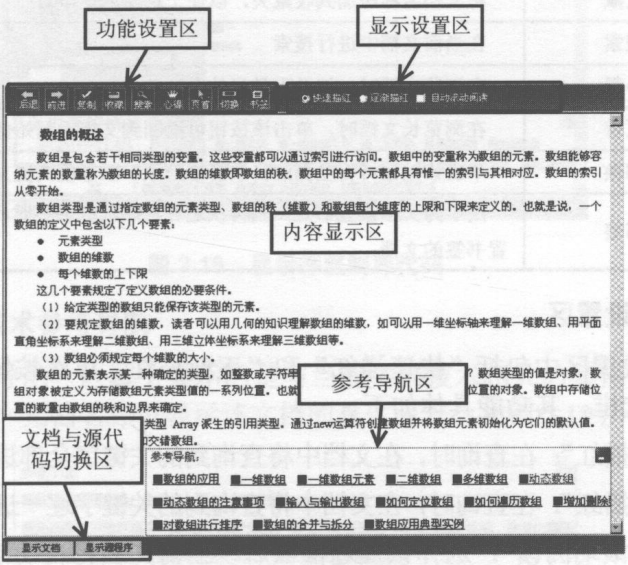


图 2.16　内容显示设置区域



提示

当打开“技术中心”、“方案中心”或者“界面中心”的文档或图片时会显示“参考导航”区域，当打开“实例中心”的文档时会显示“实例导引”区域。

1. 功能设置区

在功能设置区中包括“后退”、“前进”、“复制”、“收藏”、“搜索”、“心得”、“页首”、“切换”和“书签”9 个按钮，如图 2.17 所示。



图 2.17　功能设置区

功能设置区中各按钮的功能如表 2.1 所示。

更多资源请访问稀酷客(www.ckook.com)

表 2.1　功能设置区按钮的功能

按　钮	说　明	功　能
后退	后退	返回到上一篇浏览过的文档
前进	前进	显示单击“后退”按钮时浏览的文档
复制	复制	在显示源代码时，选择一段代码然后单击该按钮，可对代码进行复制
收藏	收藏	将文档名称添加到收藏夹，以便于再次浏览学习
搜索	搜索	在当前文档中进行搜索
心得	心得	在浏览文档时，记录下学习体会
页首	页首	在浏览长文档时，单击该按钮可返回到文档的起始位置
切换	切换	单击该按钮可将左侧的目录窗格隐藏，以方便浏览文档
书签	书签	用于为文档设置书签，当再次运行程序读取书签时，可直接打开设置书签的文档

2. 显示设置区

在显示设置区中包括“快速描红”和“逐渐描红”单选按钮和“自动滚动阅读”复选框，其功能具体如下。

- “快速描红”：在查询时，在文档中将查询到的关键字全部进行描红显示。
- “逐渐描红”：在查询时，在文档中将查询到的关键字逐一进行描红显示。
- “自动滚动阅读”：选择该复选框以后，当前文档将自动滚动显示。

3. 内容显示区

在目录中或者在查询结果中选择某个文档名称后，将在内容显示区中显示文档的具体内容。

4. 文档与源代码切换区

在文档与源代码切换区中有“显示文档”和“显示源程序”两个按钮，如果打开的文档中有程序代码，单击“显示源程序”按钮即可在源代码显示页面中单独查看程序代码。单击“显示文档”按钮又会切换到文档内容显示页面。

选择代码后，单击文档显示区工具栏中的“复制”按钮，或者单击鼠标右键在弹出的快捷菜单中选择“复制”命令对代码进行复制，如图 2.18 所示，然后可以直接粘贴到 C#环境中进行使用，或者进行二次开发。



提示

如果打开的文档中没有源代码，则“显示源程序”按钮呈灰色，不可用。

更多资源请访问稀酷客(www.ckook.com)

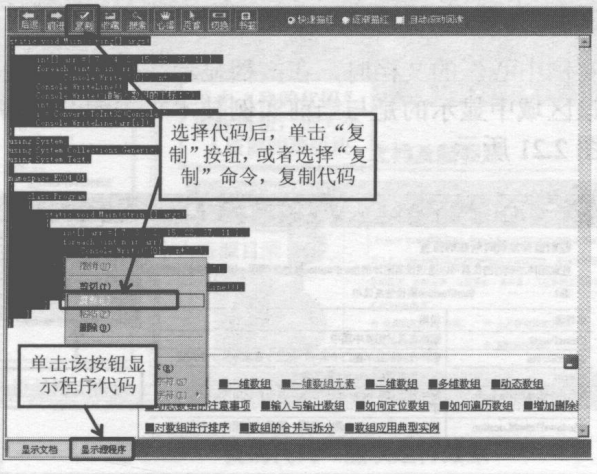


图 2.18　显示并复制源代码

5. 相关技术参考导航

在“参考导航”区域中显示的是与当前文档技术相关的其他文档名称，单击其中的任一文档名称可显示该文档的具体内容，如图 2.19 所示。

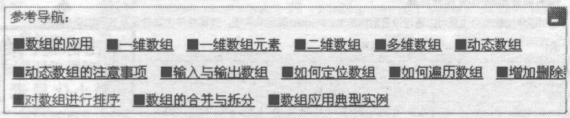


图 2.19 “参考导航”区域

单击“参考导航”区域右上角的最小化按钮，可将该区域最小化到文档显示区的最下方，如图 2.20 所示。单击最大化按钮，又可将该区域展开。

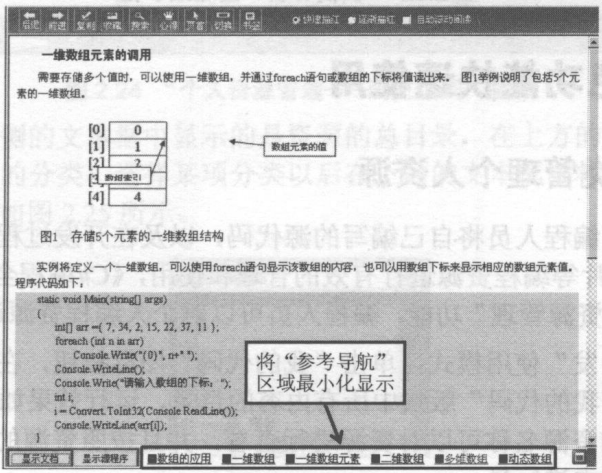


图 2.20 “参考导航”区域最小化

更多资源请访问稀酷客(www.ckook.com)

6. 相关实例参考导引

当查看“实例中心”的文档时，在文档显示区的右上角将显示“实例导引”区域，在该区域中显示的是与当前实例技术相关的其他参考实例。页面的显示效果如图 2.21 所示。

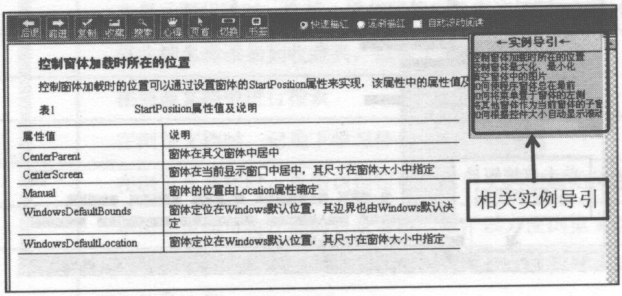


图 2.21 “实例导引”区域

在实例导引列表中单击任一实例名称，可查看其他实例的具体内容。单击该区域的标题栏，可将该区域最小化显示，如图 2.22 所示。

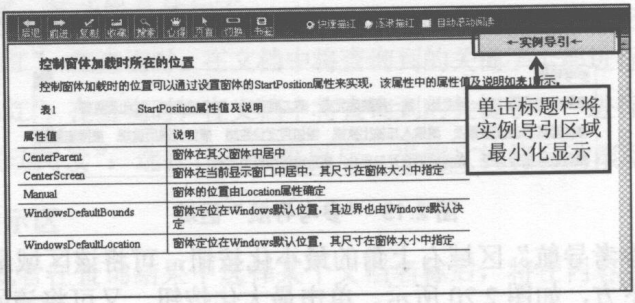


图 2.22 “实例导引”区域最小化

2.3 特色功能快速使用

2.3.1 分类管理个人资源

为了方便编程人员将自己编写的源代码，以及在开发过程中搜集到的技术、实例、图片等编程资源进行有效的管理和使用，《C#编程全能词典》软件提供了“个人资源管理”功能，编程人员可以对个人编程资源进行分类管理。

进入“开发”使用模式，单击“我的代码”图标按钮，在“目录”显示区中将显示“我的代码”版块中所有内容的目录，运行效果如图 2.23 所示。选择目录中的资源名称可以对资源进行查看，并且按照资源的文件格式在相应的运行环境中进行显示。

更多资源请访问稀酷客(www.ckook.com)

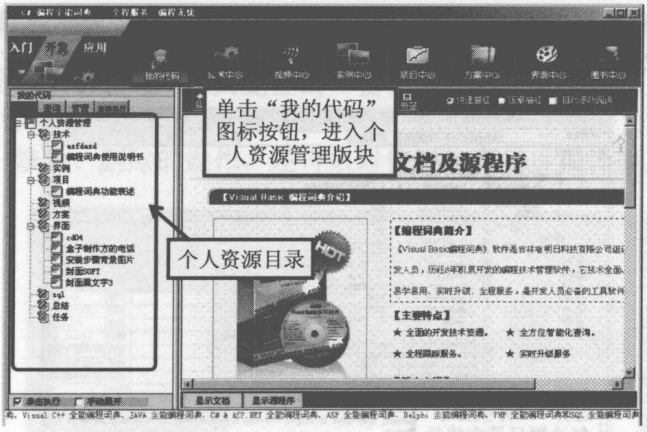


图 2.23 “我的代码”版块运行效果

1. 资源预览查询

在目录显示区中单击“管理”按钮打开“个人资源管理——浏览查询”窗口，如图 2.24 所示。



图 2.24 “个人资源管理——浏览查询”窗口

在窗口左侧的文本框中显示的是资源的总目录，在上方的下拉列表框中可以选择资源的分类，选择某项分类以后在下面的文本框中将只显示该分类的内容目录，如图 2.25 所示。

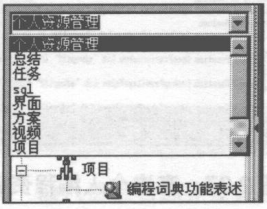


图 2.25 选择资源的分类

更多资源请访问稀酷客(www.ckook.com)

在目录文本框的下方选择“细分类显示”复选框将显示目录中的所有分类，如图 2.26 所示。

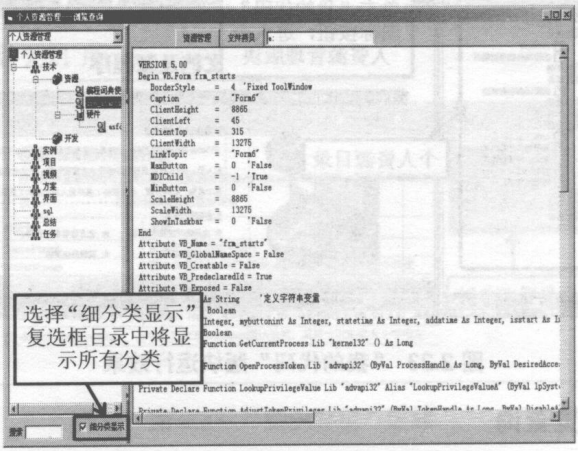


图 2.26 “细分类显示”复选框



提示

用户可以创建多级分类，从而对资源进行有效的管理。

在搜索文本框中输入查询关键字，可以在个人资源中进行快速查找，如图 2.27 所示。

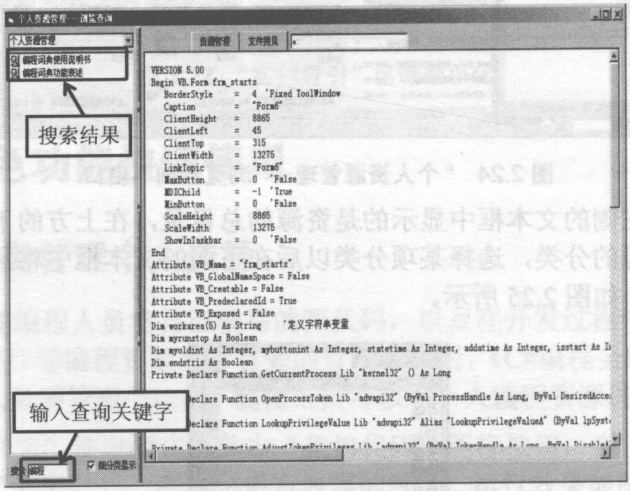


图 2.27 查询个人编程资源

更多资源请访问稀酷客(www.ckook.com)

2. 资源管理

在“个人资源管理——浏览查询”窗口中单击“资源管理”按钮，打开“个人资源管理——资源管理”窗口，如图 2.28 所示。在该窗口中可以对资源进行添加、修改、删除的操作，还可以对资源的分类进行管理。

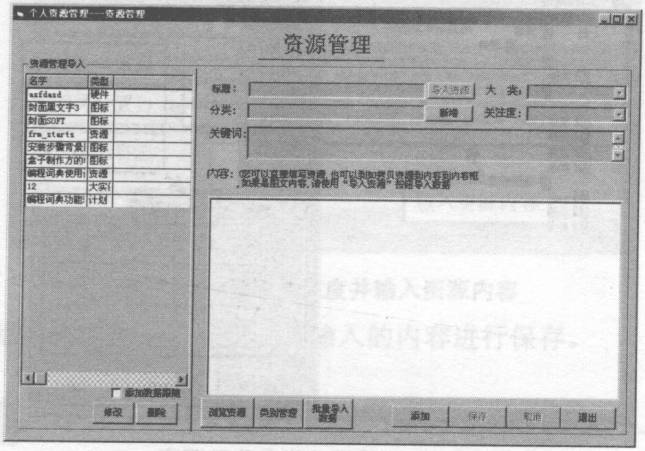


图 2.28 “个人资源管理——资源管理”窗口

添加资源的具体操作步骤如下。

1. 在窗口中单击“添加”按钮，然后在“标题”文本框中输入资源的名称，如“弹出式下拉菜单”。

2. 单击“大类”下拉列表选择该资源的所属类别，如“实例”，如图 2.29 所示。

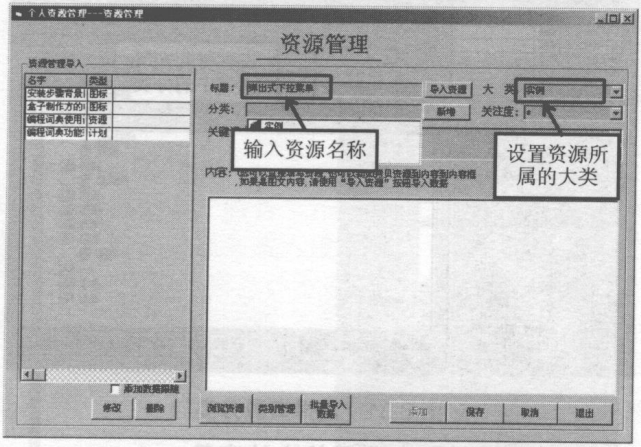


图 2.29 输入资源名称并设置所属大类

3. 选择大类以后，还需要选择在该大类中的下级分类，如果该大类没

更多资源请访问稀酷客(www.ckook.com)

有设置下级分类，则单击“分类”文本框后面的“新增”按钮，打开“类别设置”窗口。在窗口右侧的“分类名称”文本框中输入类别名称，如“菜单类”，在“分类顺序”文本框输入 1，如图 2.30 所示。

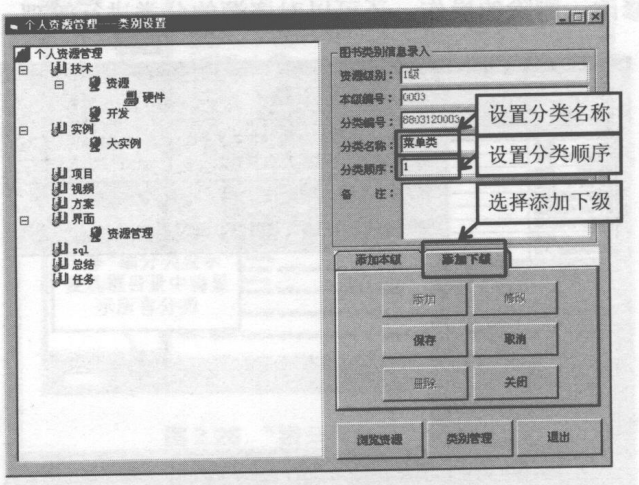


图 2.30 设置分类名称及顺序

④ 单击“保存”按钮添加分类，然后单击“关闭”按钮返回资源管理界面。

⑤ 在“分类”下拉列表框中选择刚刚新增的分类“菜单类”并双击，如图 2.31 所示。

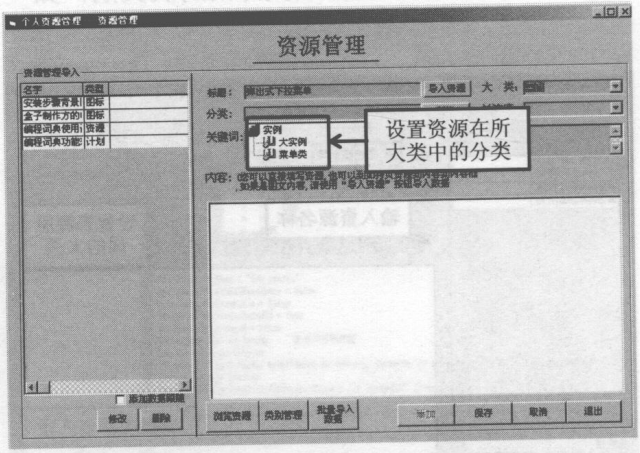


图 2.31 选择资源分类

⑥ 在“关注度”文本框中选择关注的度数。

⑦ 在“内容”文本框输入文档的内容，如图 2.32 所示。

更多资源请访问稀酷客(www.ckook.com)

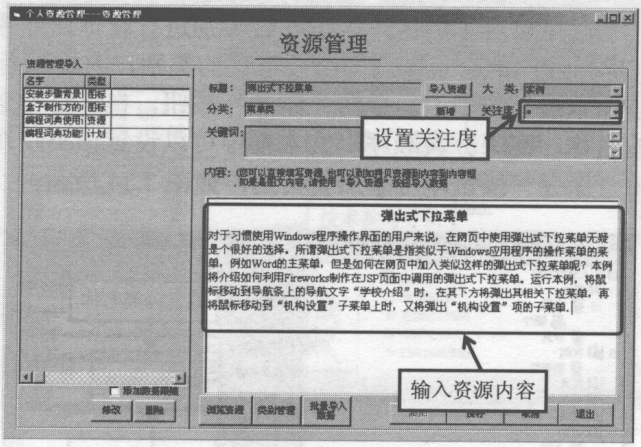


图 2.32 设置关注度并输入资源内容

⑧ 最后单击“保存”按钮，将输入的内容进行保存。

注意

通过单击“标题”文本框后面的“导入资源”按钮，可以导入已有的个人资源。

3. 类别管理

在“个人资源管理——资源管理”窗口中单击“类别管理”按钮，打开“个人资源管理——类别设置”窗口，如图 2.33 所示。在该窗口中可以对类别进行管理。

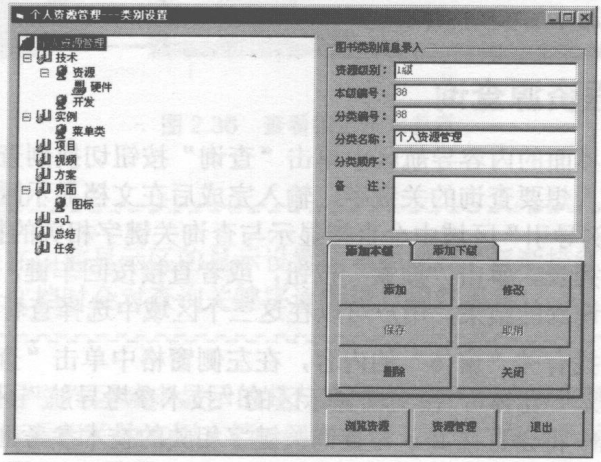


图 2.33 “个人资源管理——类别设置”窗口

更多资源请访问稀酷客(www.ckook.com)

在管理类别时可以对本级也可以对下级类别进行管理，如果要管理本级类别，首先在窗口左侧的目录中选择所要管理的类别，例如选择“任务”类，然后选择“添加本级”选项卡。单击“添加”按钮，在“分类名称”文本框中输入类别的名称，在“分类顺序”文本框中可以设置该类别在目录中的顺序，然后单击“保存”按钮即可添加新类别，如图 2.34 所示。

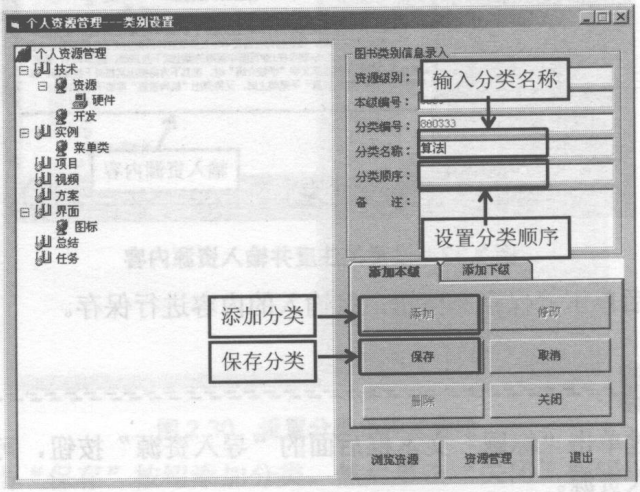


图 2.34　添加本级类别

在目录中选择某个类别后单击“修改”按钮可以对类别的名称、分类顺序等内容进行修改。

在目录中选择某个类别后单击“删除”按钮可以将该类别删除，但需要注意的是，删除类别时会同时将该类别中所保存的资源一起删除。

如果选择“添加下级”选项卡，则可以为所选择的类别添加下级类别，具体操作与添加本级类别相同。

2.3.2　编程资源查询

在程序主界面的内容导航区中单击“查询”按钮切换到查询区域，在查询文本框中输入想要查询的关键字，输入完成后在文档显示区的技术“参考导航”或“实例导引”区域中会直接显示与查询关键字相关的技术参考列表，或者实例参考列表。单击“搜索”按钮，或者直接按回车键，在下面的列表框中将显示查询到的结果，用户可以在这三个区域中选择查看。

例如，查找有关“窗体”的内容，在左侧窗格中单击“查询”按钮，在查询文本框输入“窗体”，在文档显示区的“技术参考导航”区域，或者“实例导引”区域中将会直接显示与查询关键字相关的技术参考列表，以及实例参考列表，如图 2.35 所示。

更多资源请访问稀酷客(www.ckook.com)

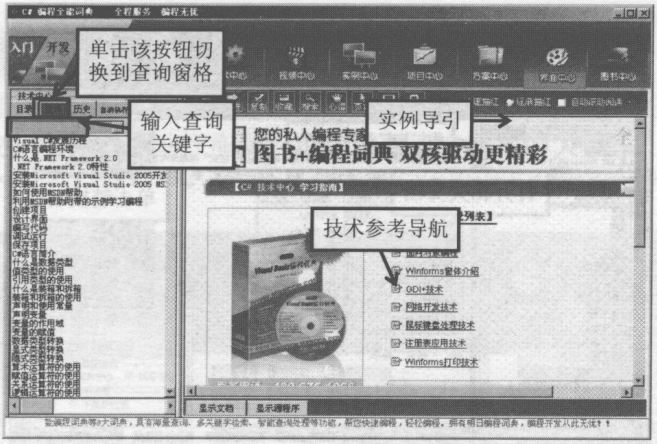


图 2.35　编程资源查询

在“技术参考导航”区域中显示的是与查询关键字相关的技术参考，单击列表中的链接名称，即可查看具体内容。例如单击“窗体设计器”，即可打开相关文档，如图 2.36 所示。

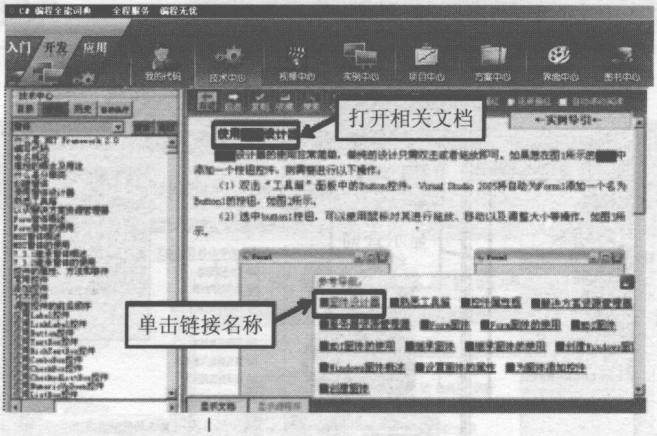


图 2.36　查看相关技术参考



提示

如果在内容显示区的显示设置区域中选择“逐渐描红”单选按钮，则在打开文档时会对查询关键字逐渐进行描红显示。

在“实例导引”区域中显示的是与查询关键字相关的实例，单击其中的链接名称，可查看相关的实例文档。例如单击“通过“格式”菜单布局窗体”，打开的文档如图 2.37 所示。

更多资源请访问稀酷客(www.ckook.com)

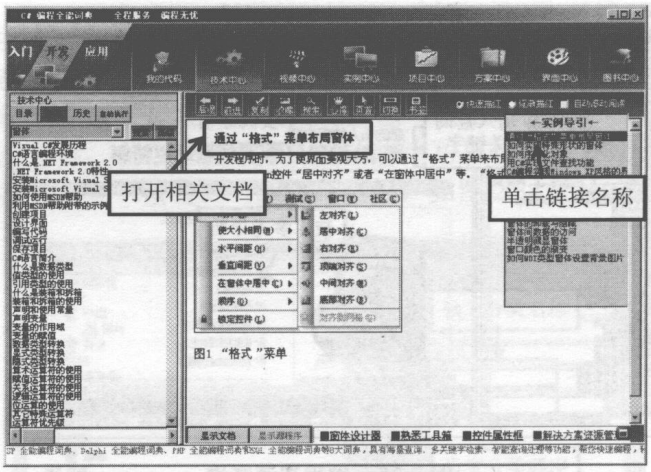


图 2.37　查看相关实例参考

输入完查询关键字以后单击“搜索”按钮，或者直接按回车键，在下面的列表框中将显示查询到的结果，如图 2.38 所示，用户也可以在其中选择查看。



图 2.38　查看查询结果

在“查询”窗格中还可以对查询的范围、字段，以及查询结果进行设置。在“查询”窗格中单击“高级”按钮，在“设置范围选择”区域中选择“所有中心”；在“查询结果包含关键字设置”区域中选择“显示位置”；在“设置查询字段”区域中选择“所有字段”，如图 2.39 所示。

单击“关闭”按钮回到“查询”窗格，单击“搜索”按钮，搜索到的结果如图 2.40 所示。

更多资源请访问稀酷客(www.ckook.com)

提示

将鼠标指针移动到“查询”窗格与文档显示区之间的分隔条上，然后按住鼠标左键进行拖动，可以调整两个区域的大小。



图 2.39　高级查询设置



图 2.40　设置后的查询结果

2.3.3　浮动窗口查询

在启动电脑时浮动查询窗口将自动运行，并隐藏在屏幕的上方。无论用户在编写文档还是开发程序，不需运行 C#编程全能词典，只需将鼠标指针移动到浮动查询窗口隐藏的位置，即可快速打开 C#编程全能词典的浮动窗口，

更多资源请访问稀酷客(www.ckook.com)

快速查询相关技术。浮动查询窗口如图 2.41 所示。



图 2.41　浮动查询窗口

查询时在“输入”文本框中输入查询关键字，在后面的下拉列表框中选择查询的中心，然后单击“搜索”按钮 搜索 开始查询。

例如，查询有关“数组”的内容。首先在“输入”文本框输入“数组”，在后面的列表框中选择“技术中心”，然后单击“搜索”按钮，查询到的结果将显示在下面的列表框中，单击想要查看的文档名称，在右侧窗格中将显示文档的具体内容，文档中查询关键字将以描红的方式显示，如图 2.42 所示。

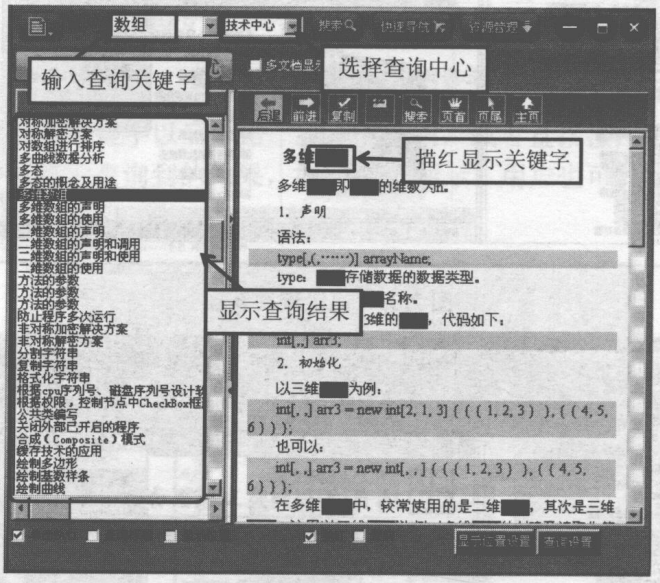


图 2.42　在浮动窗口中查询

在浮动查询窗口中，单击窗口右上角的最小化按钮，可以将窗口还原为最小化状态；单击最大化按钮，可将窗口全屏显示。单击关闭按钮，可关闭窗口。

2.3.4　收藏重要文档

用户在学习技术的过程中，如果遇到比较重要的，或者需要再次或多次浏览的技术，可以将文档的名称添加到收藏列表中。当再次使用时，无须在目录进行查找，直接打开收藏列表进行查看即可。

首先打开需要收藏的技术文档，然后单击工具栏中的“收藏”按钮，打开收藏窗口，如图 2.43 所示。

更多资源请访问稀酷客(www.ckook.com)

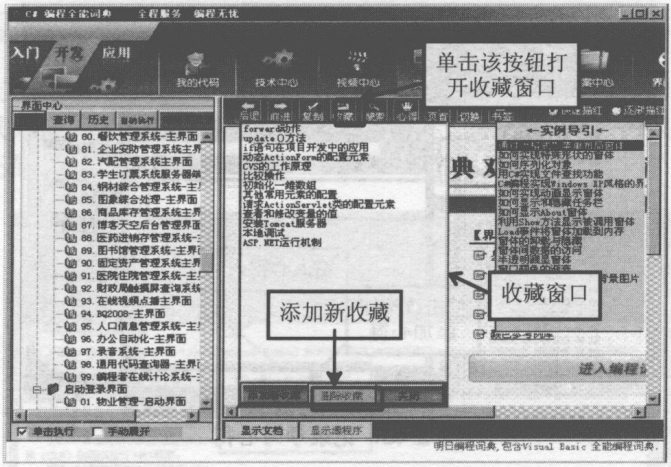


图 2.43　收藏重要文档

单击“添加新收藏”按钮，在弹出的对话框中输入文档的名称，如图 2.44 所示。最后单击“确定”按钮即可。

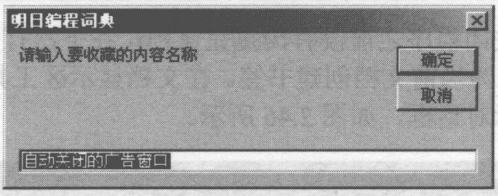


图 2.44　提示对话框

如果要浏览收藏夹中的文档，则单击“收藏”按钮，在打开的收藏窗口中双击收藏列表中的文档名称，即可将文档打开。

注意

如果选择收藏的文档名称，单击“删除收藏”按钮可以将收藏的文档删除。

2.3.5　记录学习心得

在阅读文档的过程中，用户可以把学习体会或想法等内容记录下来，日积月累，对于提高自己的编程水平会有一定的好处。在功能设置区中单击“心得”按钮 心得，打开“心得管理”窗口，在“心得内容”文本框中输入自己对这篇文档的想法和体会，然后单击“添加开发心得”按钮即可，如图 2.45 所示。

更多资源请访问稀酷客(www.ckook.com)

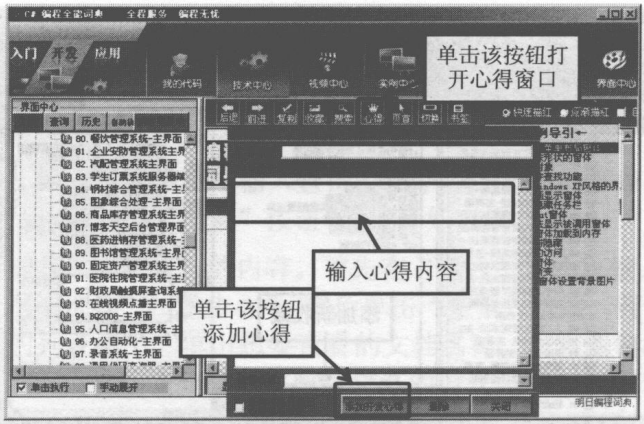


图 2.45　记录学习心得

查看心得时单击“查看心得”下拉列表框的下拉按钮，在列表中选择要查看的心得的标题，在“心得内容”文本框中即可显示该文档的心得内容。

2.3.6　为文档设置书签

如果用户在关闭程序之前没有浏览完文档，在下次打开程序时还想要继续浏览，这时可以为当前文档创建书签。在文档显示区工具栏中单击“书签”按钮，将弹出提示对话框，如图 2.46 所示。

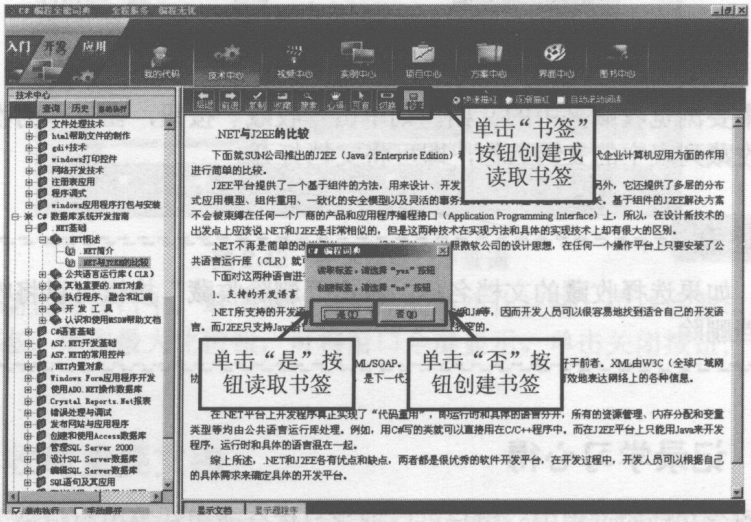


图 2.46　为文档设置书签

首先单击“否”按钮为文档设置书签；当再次运行程序时，单击“书签”按钮，在弹出的提示对话框中单击“是”按钮读取书签，即可直接将上次未浏览完的文档打开。

更多资源请访问稀酷客(www.ckook.com)

2.3.7 切换内容显示区

单击文档显示区工具栏中的“切换”按钮 切换，可以将左侧的目录窗格隐藏起来，以方便浏览文档，效果如图 2.47 所示。再次单击“切换”按钮，又可将界面还原。

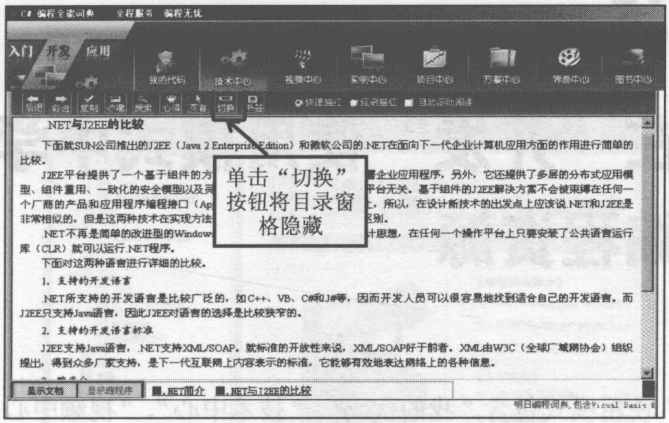


图 2.47 隐藏目录窗格

更多资源请访问稀酷客(www.ckook.com)

第 3 章

如何在“开发”使用模式中学习和使用编程资源

“开发”使用模式包含“我的代码”、“技术中心”、“视频中心”、“实例中心”、“方案中心”、“项目中心”、“界面中心”和“图书中心”8 个版块，其中包含 **1000 多**篇技术文档；**1500 多个**实例、技巧；**30** 个小时的技术和项目的视频讲解；**25** 套实用项目及 **5000** 张风格独特的界面设计方案，内容覆盖编程所需的所有资源，通过“我的代码”版块，用户可以对自己的编程资源进行分类管理。

“开发”模式的内容结构图如图 3.1 所示。

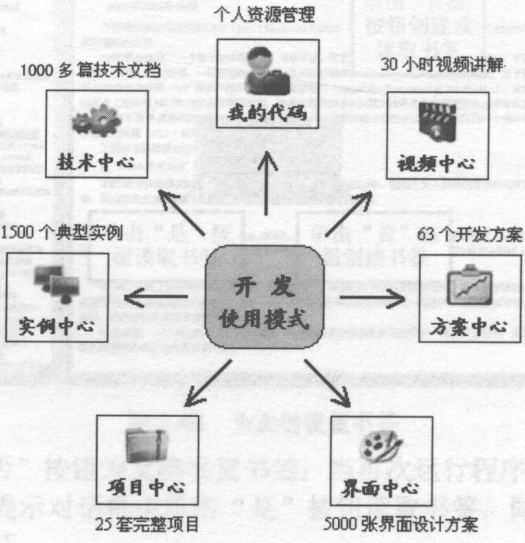


图 3.1 “开发”模式内容结构图

更多资源请访问稀酷客(www.ckook.com)

3.1 如何分类管理个人编程资源

“我的代码”版块主要用于分类管理用户的个人编程资源，在功能导航区中单击“我的代码”图标按钮，在目录显示区中将显示“我的代码”版块中所有内容的目录，运行效果如图 3.2 所示。

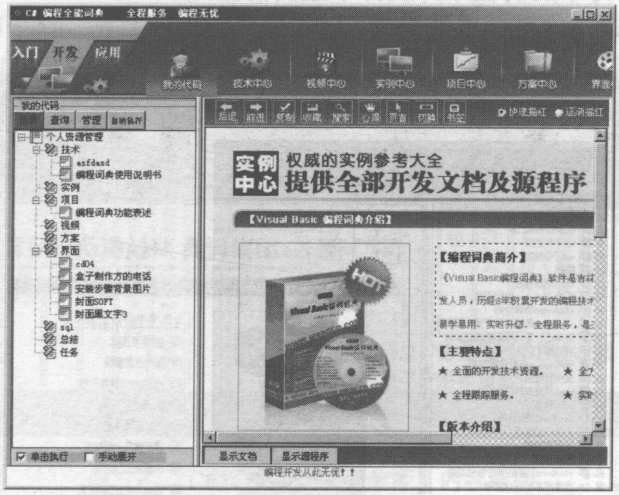


图 3.2 “我的代码”版块运行效果

在目录中选择资源名称可以打开资源查看，并可以根据资源的文件格式，在相应的运行环境中打开。

在目录显示区中单击“管理”按钮，将打开“个人资源管理——浏览查询”窗口，如图 3.3 所示，在该窗口中可以对资源进行添加、修改、删除操作。

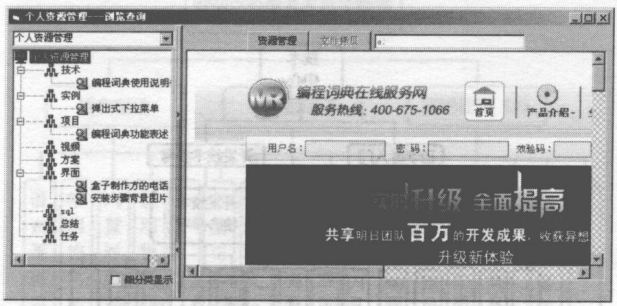


图 3.3 “个人资源管理——浏览查询”窗口



提示

具体操作方法请参见 2.3.1 节的内容。

更多资源请访问稀酷客(www.ckook.com)

3.2 如何学习浏览编程技术资源

学好编程技术是进行程序开发的基础，只有全面并熟练掌握各项编程技术才能开发出好的程序。在《C#编程全能词典》的“技术中心”为编程人员提供了在程序开发过程中所需的最全面、最具权威性的技术文档，通过这部分的学习，将为今后的程序开发打下坚实的基础。

界面运行效果如图 3.4 所示。

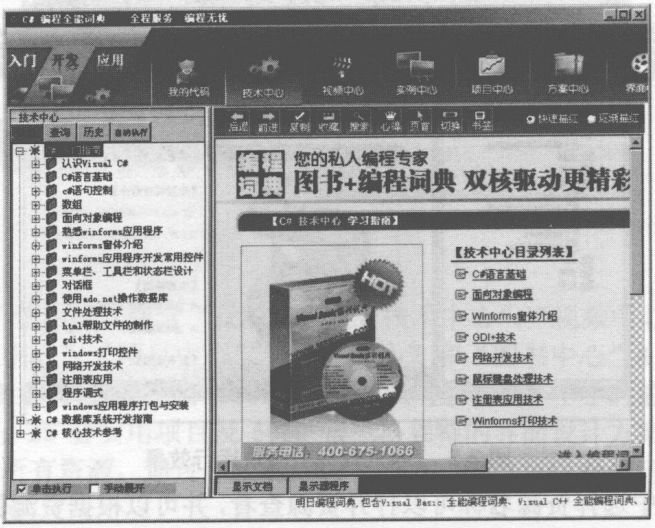


图 3.4 “技术中心”版块界面运行效果

“技术中心”版块中提供了上千篇技术文档，内容主要包括程序设计自学参考、数据库系统开发自学参考、开发技术核心参考等内容，其内容结构如图 3.5 所示。

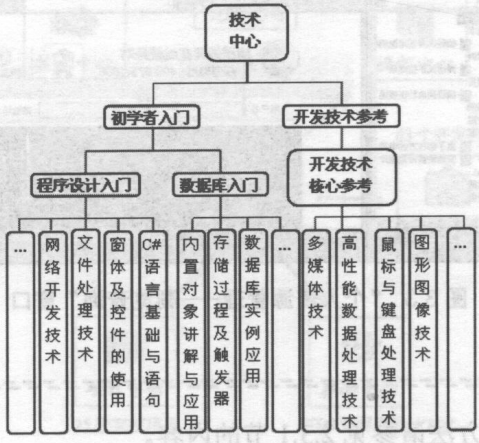


图 3.5 “技术中心”内容结构

更多资源请访问稀酷客(www.ckook.com)

用户在学习“技术中心”中的内容时，可以通过目录进行学习。在“开发”使用模式的功能导航区中单击“技术中心”图标按钮，进入技术中心界面。然后在目录显示区中展开相应的节点。

1. C#程序设计自学参考

“C#程序设计自学参考”囊括了使用C#进行程序设计的全部知识，包括环境搭建、语言基础、算法和流程控制语句、网络开发等技术。同时，配合实用性的示例作为辅助，帮助读者举一反三，速学速用。内容结构如图3.6所示。

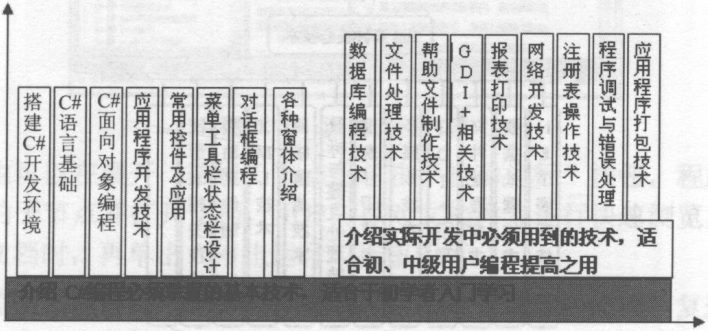


图3.6 “C#程序设计自学参考”内容结构

2. 数据库系统开发自学手册

“数据库系统开发自学手册”主要介绍C#语言基础知识、报表打印、数据库管理、SQL语句及其应用、存储过程及视图等技术。提供精炼有效的代码来巩固和扩充用户的编程知识，提高编程水平。内容结构如图3.7所示。

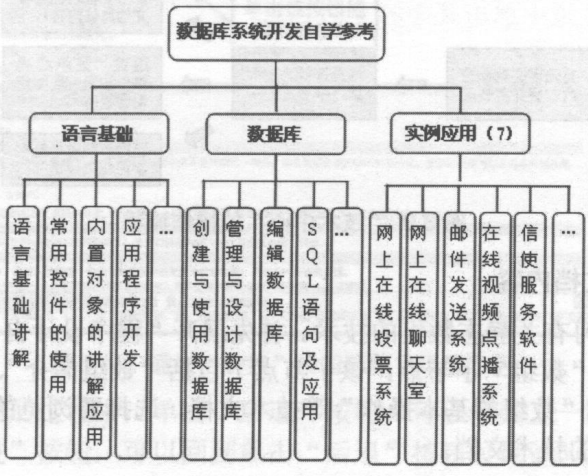


图3.7 “数据库系统开发自学手册”内容结构

更多资源请访问稀酷客(www.ckook.com)

3. 开发技术核心参考

“开发技术核心参考”提供了高性能数据处理、文件处理、多媒体、水晶报表、XML 语言及打印等技术。用户在编程过程中可参考工程项目开发思路、方法和经验，轻松解决项目开发中遇到的问题。内容结构如图 3.8 所示。

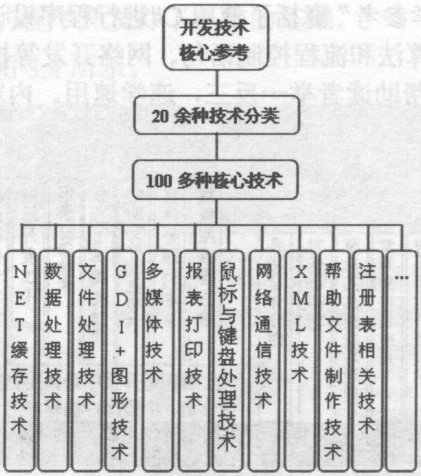


图 3.8 “开发技术核心参考”内容结构

用户可以按照目录结构依次展开子节点，每个子节点都包含了与子节点名称相关的若干个技术文档，单击想要学习的技术文档名称，即可打开相应的技术文档，在界面右侧的内容显示区中将会查看到文档的具体内容。

“技术中心”的操作流程如图 3.9 所示。

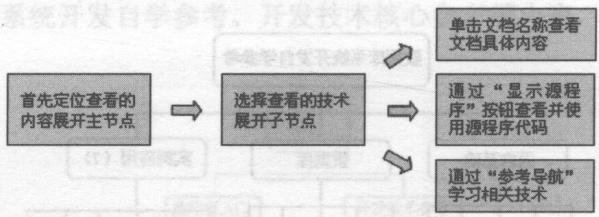


图 3.9 “技术中心”的操作流程

1. 浏览文档内容

例如，学习有关静态数组的技术。首先展开“程序设计自学参考”节点，再展开其中的“数组”子节点，该子节点下包括“数组简介”、“静态数组”、“动态数组”和“数组的基本操作”等技术文档，选择要浏览的技术文档名称即可打开相应的技术文档。

展开“静态数组”节点，然后单击“一维数组的声明”技术文档名称，在右侧的内容显示区中将显示文档的具体内容，运行效果如图 3.10 所示。

更多资源请访问稀酷客(www.ckook.com)

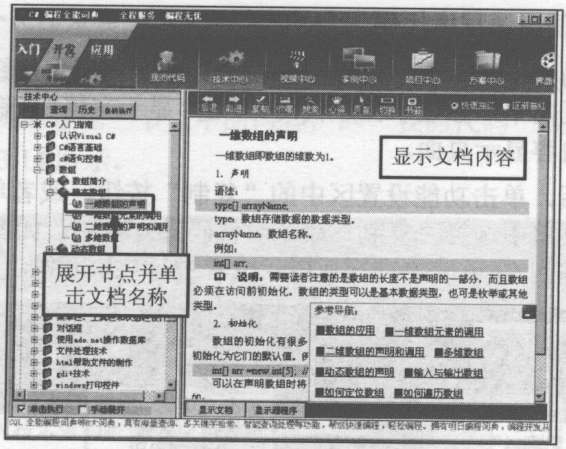


图 3.10　浏览文档内容

用户也可以在展开“数组”节点后单击“自动执行”按钮，程序将自动展开其中的子节点并打开文档，用户可以对文档内容进行快速浏览，找到想要浏览的文档时，再单击“停止运行”按钮即可。

在浏览文档内容时，选择功能设置区中的“自动滚动阅读”复选框，文档显示区将自动滚屏显示，不需要用户通过滚动条或鼠标进行向下翻页。

单击功能设置区中的“切换”按钮，可以将左侧的目录窗格隐藏，以方便浏览文档，效果如图 3.11 所示。

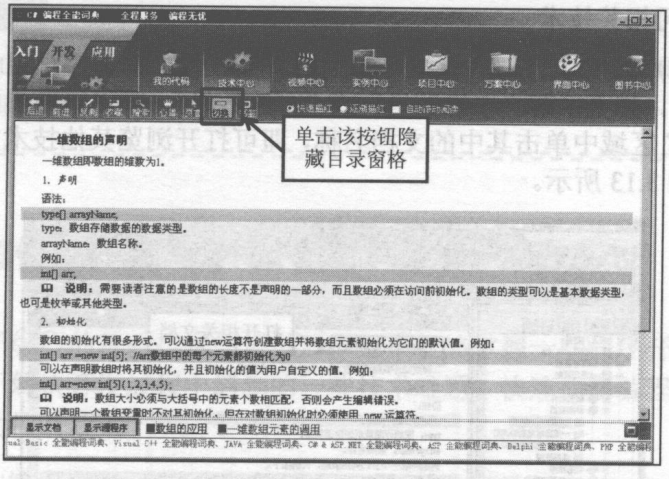


图 3.11　通过“切换”按钮隐藏目录显示区

在浏览文档的过程中，单击“后退”按钮，可以回到上一篇浏览过的文档；单击“前进”按钮，可以回到单击“后退”按钮时浏览的文档；如果文档内容较长，单击“页首”按钮，可以快速回到文档的起始位置。

更多资源请访问稀酷客(www.ckook.com)

2. 查看并使用源程序代码

如果打开的文档中有程序代码，单击“显示源程序”按钮 显示源程序 将在源代码显示页面中单独显示程序代码。单击“显示文档”按钮 显示文档 又会切换到文档内容显示页面。

选择代码后，单击功能设置区中的“复制”按钮，或者单击鼠标右键，在弹出的快捷菜单中选择“复制”命令对代码进行复制，如图 3.12 所示，然后可以直接粘贴到 C#环境中进行使用，或者进行二次开发。

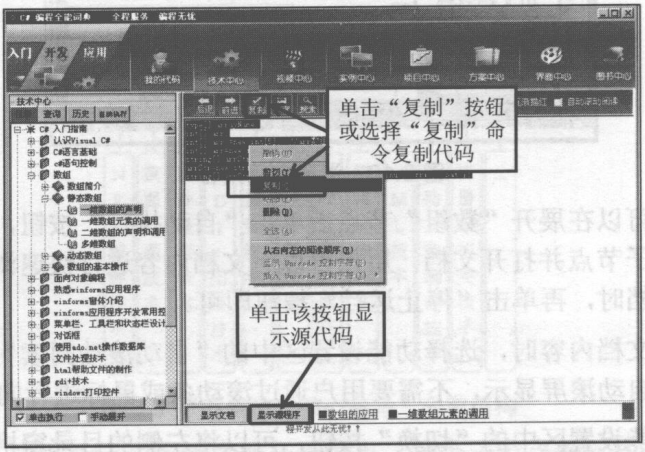


图 3.12 查看并复制源代码

3. 学习相关技术

如果用户想要学习与当前技术相关的其他技术，例如在学习“一维数组的声明”时，想要了解“二维数组”或“多维数组”，则可以在界面右下方的“参考导航”区域中单击其中的文档名称，即可打开浏览其他技术文档的具体内容，如图 3.13 所示。

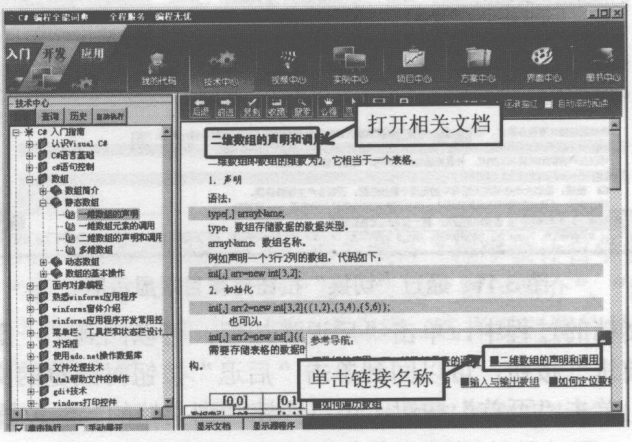


图 3.13 查看相关技术文档

更多资源请访问稀酷客(www.ckook.com)

4. 管理个人技术资源

在“技术中心”中还可以对用户的个人技术资源进行查看，目录显示区中“个人技术资源管理”节点中的内容，即“我的代码”版块中“技术”类的内容，如图 3.14 所示。



图 3.14　查看个人技术资源



提示

如果需要对个人技术资源进行添加、修改、删除等操作可以在“我的代码”版块中进行设置。

3.3　如何学习浏览应用实例资源

掌握了编程的基础技术知识以后，如果能够掌握一些程序开发的经验和技巧，对于解决开发过程中遇到的一些技术难题，快速提高编程水平，可以起到事半功倍的效果。

在《C#编程全能词典》的“实例中心”版块中精选了 **1500** 多个典型实例，内容覆盖了开发中的热点和关键问题。所选实例均源于实际项目的开发，实例讲解清晰、透彻。“实例中心”界面运行效果如图 3.15 所示。

更多资源请访问稀酷客(www.ckook.com)

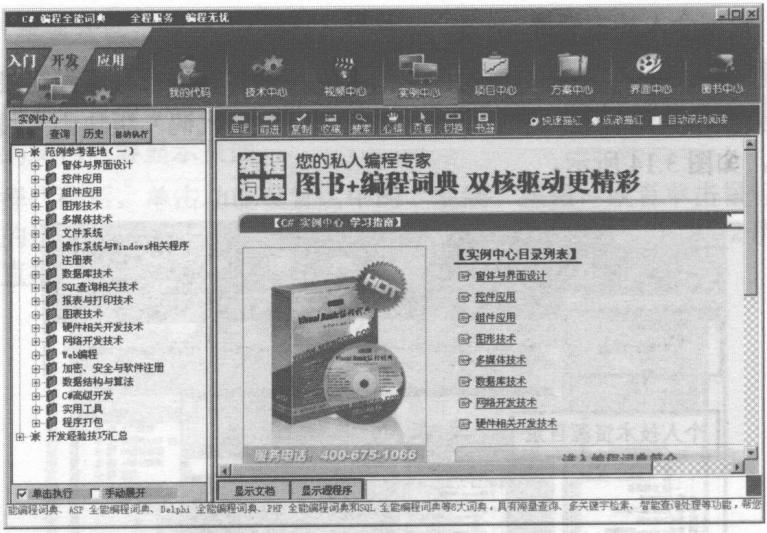


图 3.15 “实例中心”版块界面运行效果

在“实例中心”版块中提供了程序开发过程中经常用到的 **500** 多个各种典型实例和 **1000** 多个经验技巧，用户不仅可以快速学习实例，又可以通过实例解决开发难题，真正快学、快用。实例中心的内容结构如图 3.16 所示。

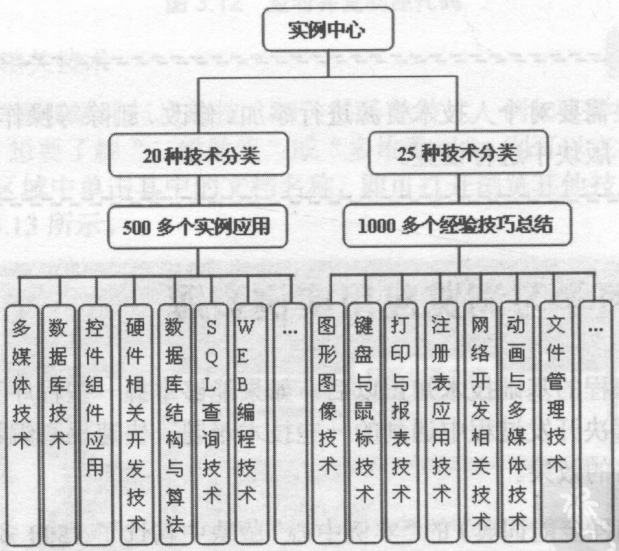


图 3.16 “实例中心”内容结构

1. 浏览实例文档

用户可以通过目录浏览实例文档，首先按照目录结构依次展开节点，每

更多资源请访问稀酷客(www.ckook.com)

个子节点都包含了与子节点名称相关的若干个实例文档，单击想要学习的实例文档名称，即可打开相应的实例文档，在界面右侧的内容显示区中将会查看到文档的具体内容。

“实例中心”的操作流程如图 3.17 所示。

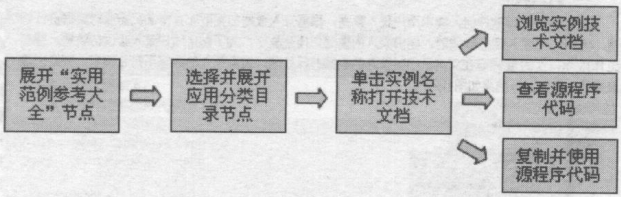


图 3.17 “实例中心”的操作流程

例如，查看“控件应用”方面的实例，则展开“控件应用”节点，然后选择其中的“只允许输入数字的 TextBox 控件”文档，运行效果如图 3.18 所示。

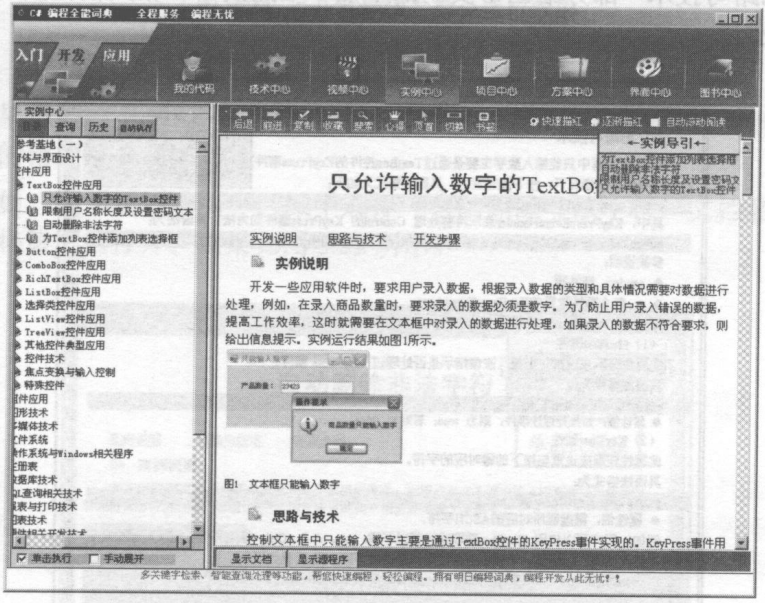


图 3.18 查看实例

在实例的讲解上，文档采用了统一的编排方式，按照实例说明、技术要点、开发步骤三个方面进行讲解。同时为了阅读方便，文档中实例名称的下方提供了超链接，浏览文档时无须使用滚动条，直接选择文档名称即可快速定位到文档位置。

“实例说明”部分以图文结合的方式给出了实例的功能和运行效果图，如图 3.19 所示。

更多资源请访问稀酷客(www.ckook.com)

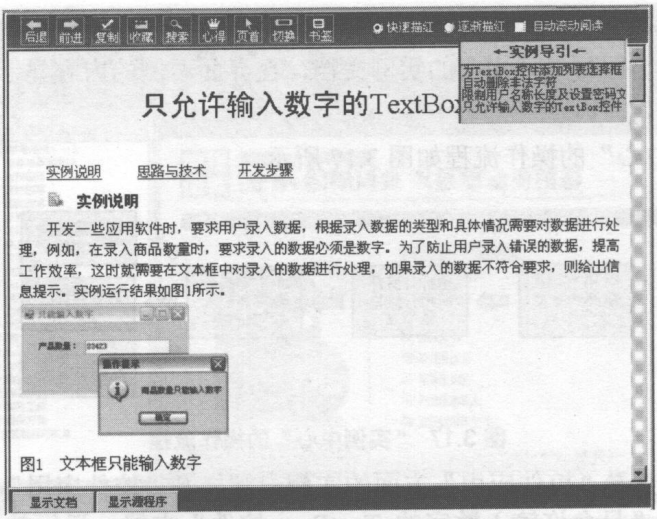


图 3.19 “实例说明”部分运行效果

“思路与技术”部分给出了实例的重点、难点技术和相关编程技巧，如图 3.20 所示。

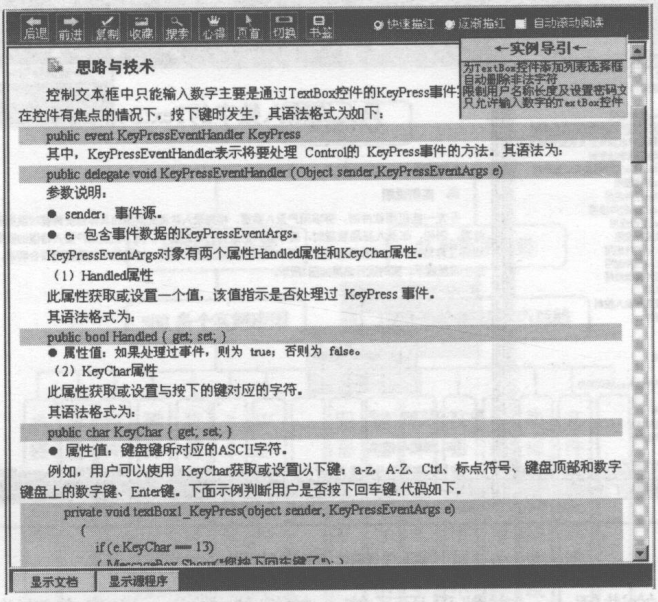


图 3.20 “思路与技术”部分运行效果

“开发步骤”部分给出了实例的设计过程和主要程序代码，如图 3.21 所示。

如果需要在实例文档中查询相关技术，可以通过文档上方的“搜索”按钮进行操作。

更多资源请访问稀酷客(www.ckook.com)

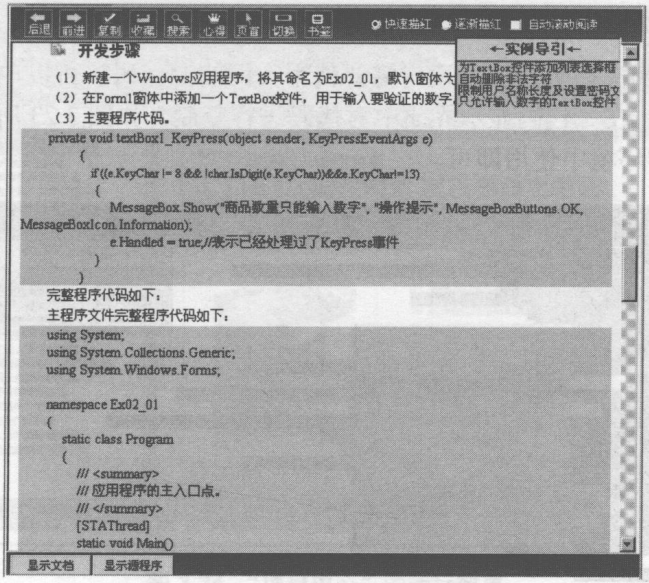


图 3.21 “开发步骤”部分运行效果

2. 浏览同类别实例文档

如果想要浏览与当前实例相关的其他实例文档，可以使用“实例导引”区域中的相关链接。

例如，在“实例导引”区域中选择“自动删除非法字符”文档名称，即可快速将该文档打开。效果如图 3.22 所示。

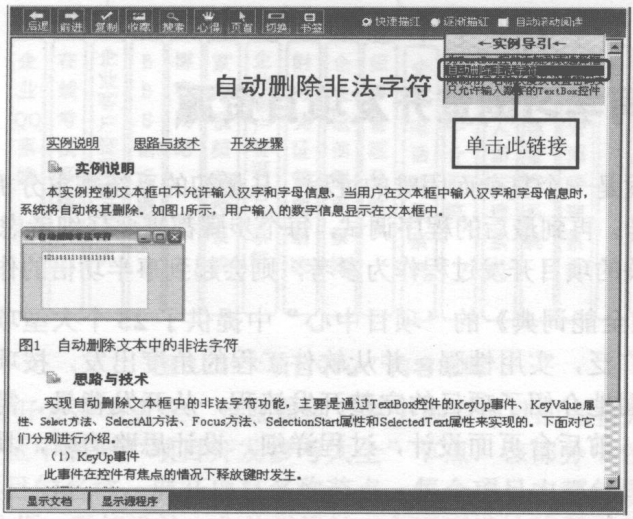


图 3.22 浏览同类别文档

更多资源请访问稀酷客(www.ckook.com)

3. 复制源码，进行二次开发

浏览实例文档时，如果读者需要使用实例的源代码，可以单击“显示源程序”按钮，在源代码显示界面中对代码进行复制，如图 3.23 所示。然后直接粘贴到 C#环境中使用即可。

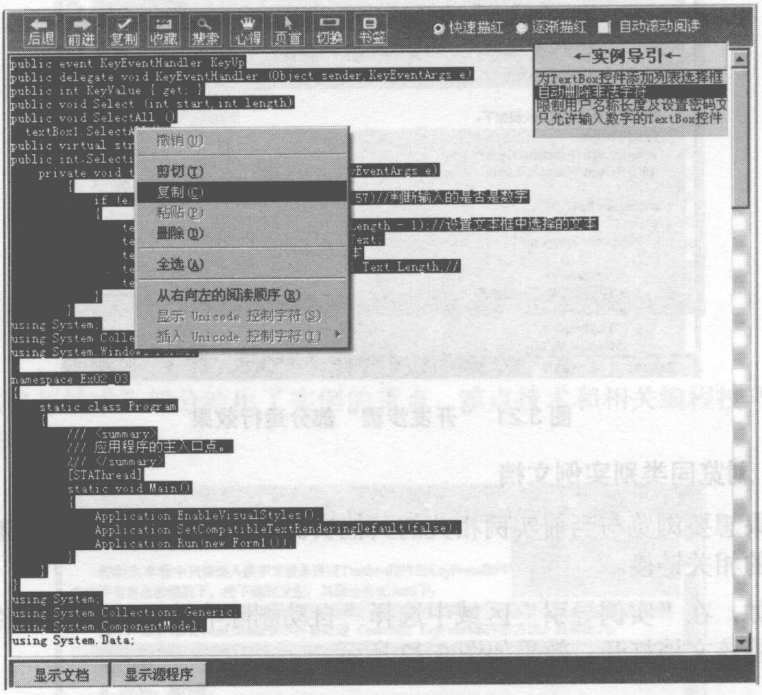


图 3.23　显示并复制源代码

3.4　如何学习浏览开发项目资源

项目开发是一个复杂而困难的过程，从最初的系统需求分析，到各个页面功能的设计，再到最后的程序调试，每个步骤都需要仔细考虑，缜密规划。如果能有相关的项目开发过程作为参考，则会起到事半功倍的作用。

《C#编程全能词典》的“项目中心”中提供了 **25** 个大型项目案例，项目涉及行业广泛，实用性强，并从软件工程的角度出发，按项目的开发顺序系统、全面地介绍了项目的完整开发流程，从开发背景、需求分析到系统功能分析、前后台页面设计，过程详细，设计思路清晰，具有较好的参考价值。

“项目中心”的运行效果如图 3.24 所示。

更多资源请访问稀酷客(www.ckook.com)

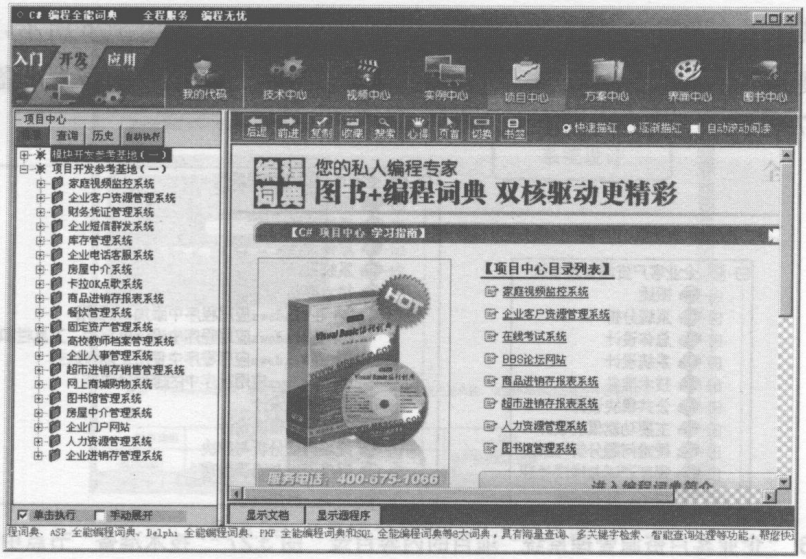


图 3.24 “项目中心”的运行效果

“项目中心”中分别介绍了 **20** 个大型项目的开发过程参考和 **5** 个典型模块的开发过程参考。其内容结构如图 3.25 所示。

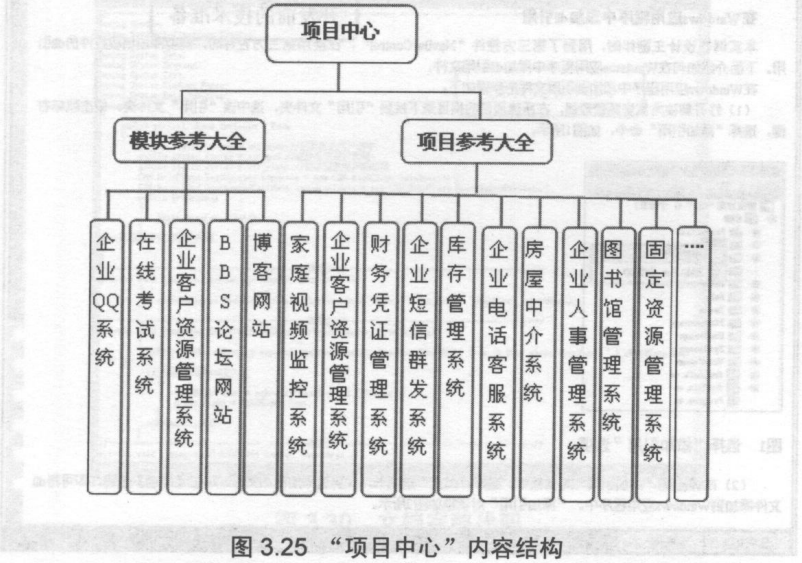


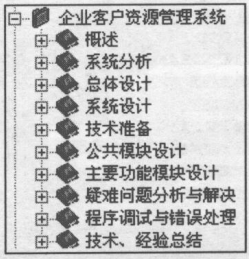
图 3.25 “项目中心”内容结构

在“项目开发参考大全”节点提供了 **5** 个典型项目的完整开发过程。在“目录”显示区中展开“项目开发参考大全”节点，该部分中包含了不同行业的 **20** 个项目的名称。展开项目名称节点，每个项目中都包含项目的开发背景、需求分析、系统设计、公共类设计、主页面设计等项目开发每个过程的内容。

更多资源请访问稀酷客(www.ckook.com)

其中“企业客户资源管理系统”项目的内容目录，如图 3.26 所示。

展开“技术准备”节点，这部分主要介绍了程序开发前的技术准备，如图 3.27 所示。



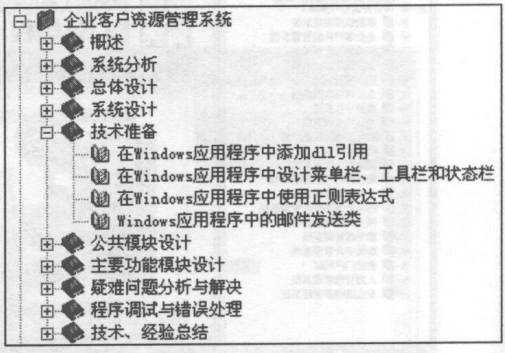


图 3.26 “企业客户资源管理系统”项目的内容目录　图 3.27 “技术准备”节点中的内容

选择“在 Windows 应用程序中添加 dll 引用”技术文档，运行结果如图 3.28 所示。

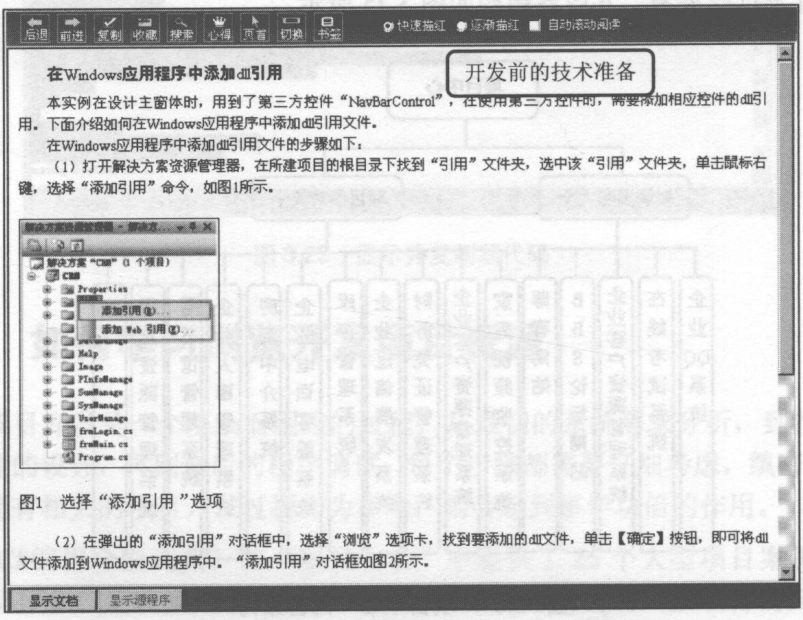


图 3.28 “开发前的技术准备”文档界面

展开“主要功能模块设计”节点，这部分主要介绍了主要功能模块的设计实现过程。选择“系统登录设计”技术文档可浏览文件架构设计图，效果如图 3.29 所示。

更多资源请访问稀酷客(www.ckook.com)

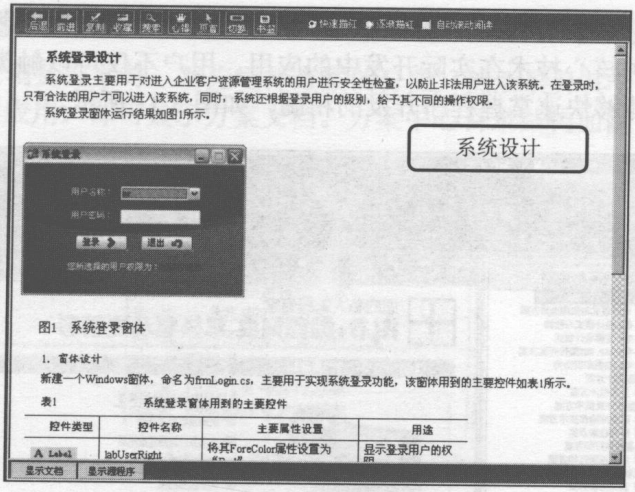


图 3.29 “系统登录设计”文档界面

如果单击界面下方的“显示源程序”按钮，可在源程序界面浏览该模块的完整代码，如图 3.30 所示。用户可以将代码直接复制至开发环境中进行二次开发。

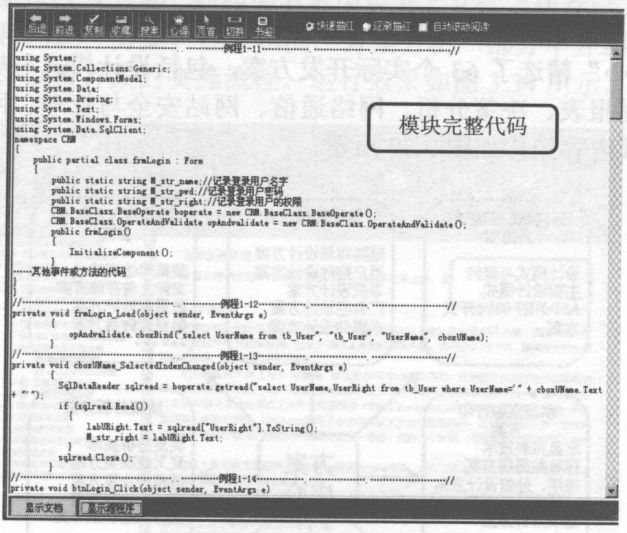


图 3.30 文档完整代码

3.5 如何学习浏览应用方案资源

在学习任何一种编程语言的过程中，仅仅学习各种编程技术的基础知识是不够的，关键在于如何将编程技术真正应用到实际当中去，只有熟练掌握各项编程技术的各种应用方案，才能开发出真正好的程序。

更多资源请访问稀酷客(www.ckook.com)

《C#编程全能词典》的“方案中心”从编程必须掌握的核心技术入手，深入讲解了各种核心技术在实际开发中的应用。用户不仅可以触类旁通，学以致用，而且能够快速掌握程序开发的精髓，如图 3.31 所示。



图 3.31 “方案中心”运行效果

“方案中心”精选了 63 个实际开发方案，包括设计模式与架构选择、页面设计、水晶报表、决策分析、网络通信、网站安全与管理、网站编译与发布等方案。其内容结构如图 3.32 所示。

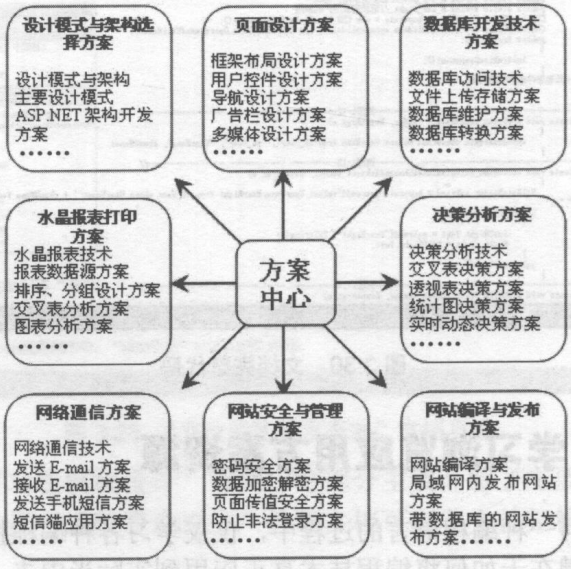


图 3.32 “方案中心”内容结构

更多资源请访问稀酷客(www.ckook.com)

1. 方案实施分析

通过目录浏览方案时，首先选择方案的分类，然后展开其节点，最后选择具体的方案应用，即可显示方案文档。方案中心目录结构如图 3.33 所示。

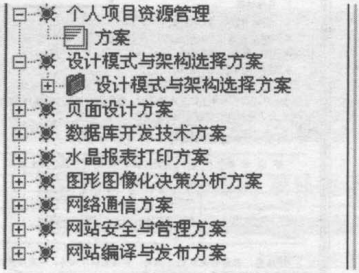


图 3.33 “方案中心”目录结构

每个应用方案按照方案分析→实施过程→补充说明的方式详细介绍了C#技术在方案中的实际应用。

例如：浏览“数据库开发技术方案”中的“存储文件路径”方案。

依次展开“数据库开发技术方案”/“数据库操作技术方案”节点，在所有与“数据库操作技术方案”的相关方案中展开“文件上传存储方案”，然后选择“存储文件路径”方案。在文档的“方案分析”部分中介绍了实现方案所采用的技术，以及方案的实施流程。运行效果如图 3.34 所示。

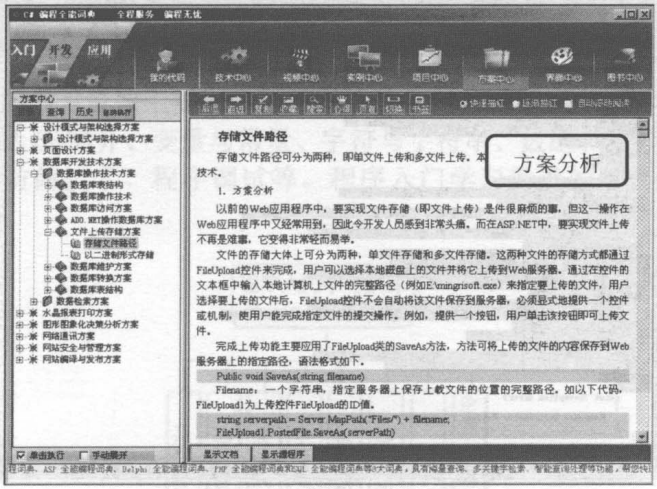


图 3.34 “方案分析”运行界面

2. 方案实施过程

在“实施过程”部分中详细介绍了方案的完整实施过程，包括方案所实现的效果，以及其中关键代码的编写过程，如图 3.35 所示。

更多资源请访问稀酷客(www.ckook.com)

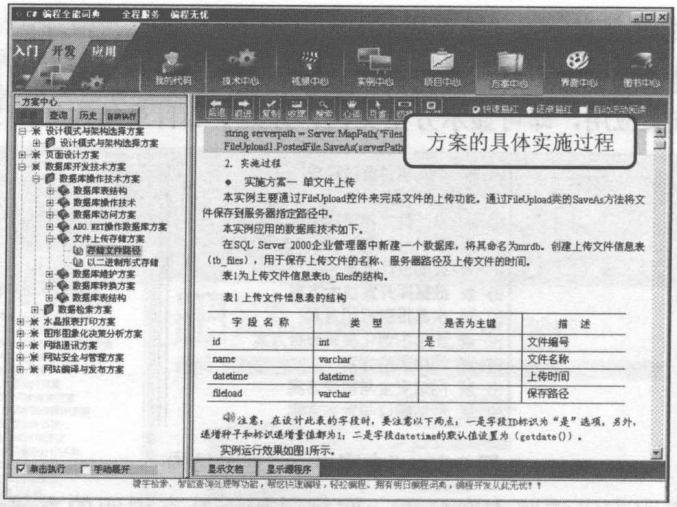


图 3.35 “方案实施过程”运行界面

3. 复制使用源代码

单击界面下方的“显示源程序”按钮可以查看该方案的完整程序代码，用户可以将这些代码直接复制并粘贴至开发环境中进行二次开发，如图 3.36 所示。

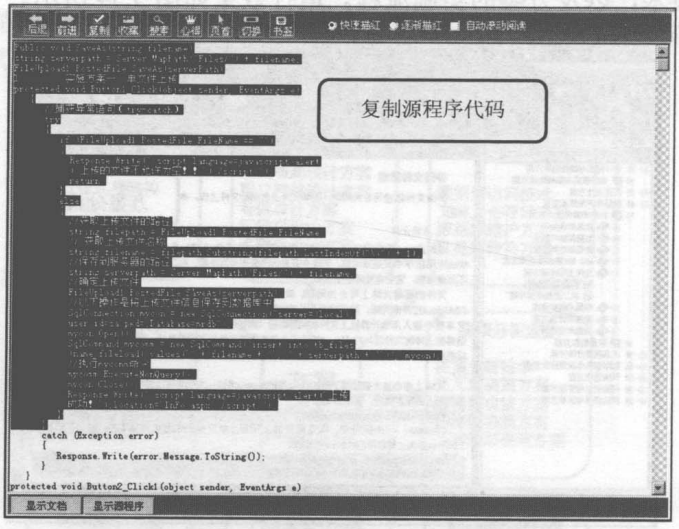


图 3.36 “复制代码”界面

3.6 观看入门及开发视频资源

“视频中心”是初学编程的爱好者快速入门的平台，30 个小时录像包括

更多资源请访问稀酷客(www.ckook.com)

安装配置录像、项目开发全过程录像和核心技术应用录像，可以让零基础编程人员快速掌握编程技巧、全方位掌握项目开发流程。

“视频中心”界面运行结果如图 3.37 所示。

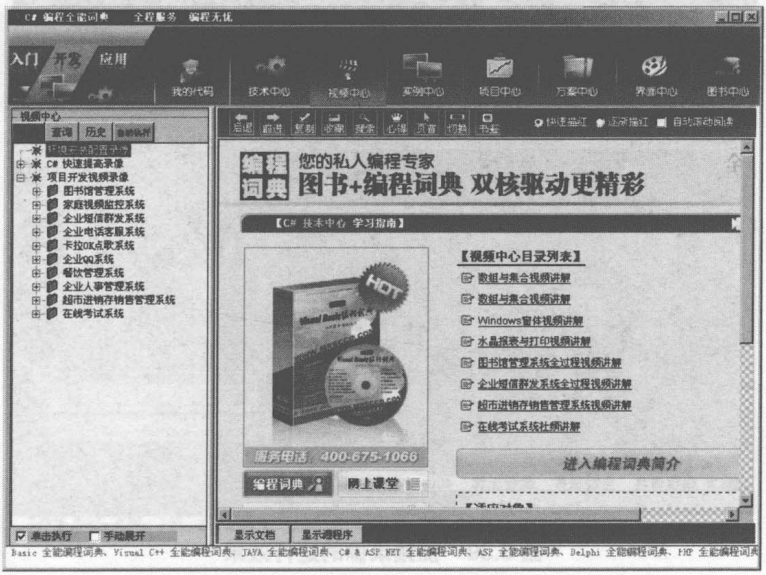


图 3.37 “视频中心”界面运行结果

3.6.1 程序入门学习录像

程序入门学习录像主要引导初学者快速入门，增强进一步学习的信心。内容主要包括 C#简介、变量与常量、字符与字符串、数组和集合、属性和方法、数据库访问技术、程序调试等。程序入门学习录像的基本内容结构如图 3.38 所示。

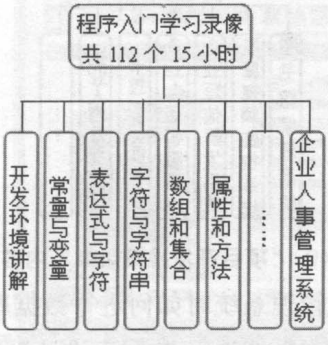


图 3.38 “程序入门学习录像”内容结构

初学者可以通过目录有选择性的试听录像，如展开“程序入门学习录

更多资源请访问稀酷客(www.ckook.com)

像”，选择“GDI+图形图像技术”/“GDI+绘图的应用”录像，录像图如图3.39所示。



图 3.39 视频录像的运行效果

3.6.2 项目开发视频录像

项目开发视频录像主要以项目的形式，从前言、功能概述、数据库设计到主窗体设计都配备了视频讲解。通过视频引导读者如何开发一个项目。其基本内容结构如图3.40所示。

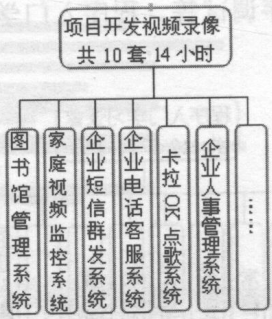


图 3.40 “项目开发视频录像”内容结构

如在开发一个图书馆管理系统时如何进行数据库的设计？

通过项目“图书馆管理系统”/“数据库设计”的视频讲解，相信编程人员可以轻而易举的学会数据库的设计，并能够举一反三，自行设计其他类似的数据库。“数据库设计”的视频效果如图3.41所示。

更多资源请访问稀酷客(www.ckook.com)

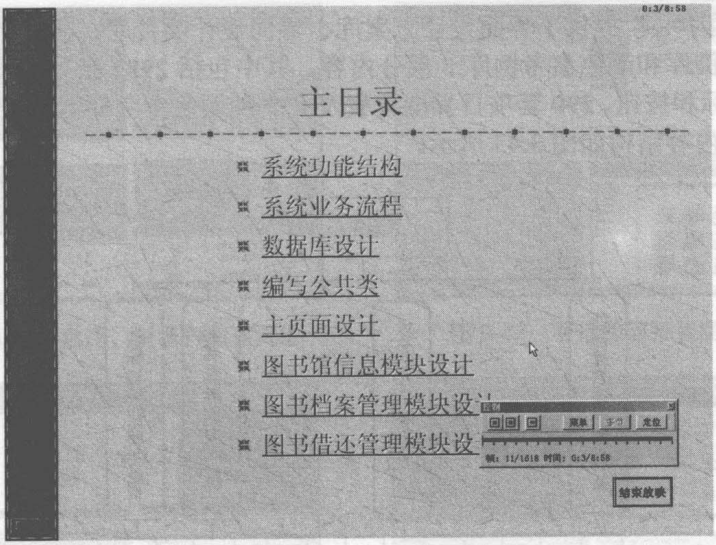


图 3.41　项目开发视频录像运行效果

3.7　浏览及使用界面素材资源

“界面中心”提供了应用类及网页类程序的上千个界面设计素材，从图标、按钮到主界面、登录界面等各种程序界面，从实例到项目，几乎囊括了程序开发所需要的各种界面，以及色彩知识、配色要点及技巧的运用。效果如图 3.42 所示。

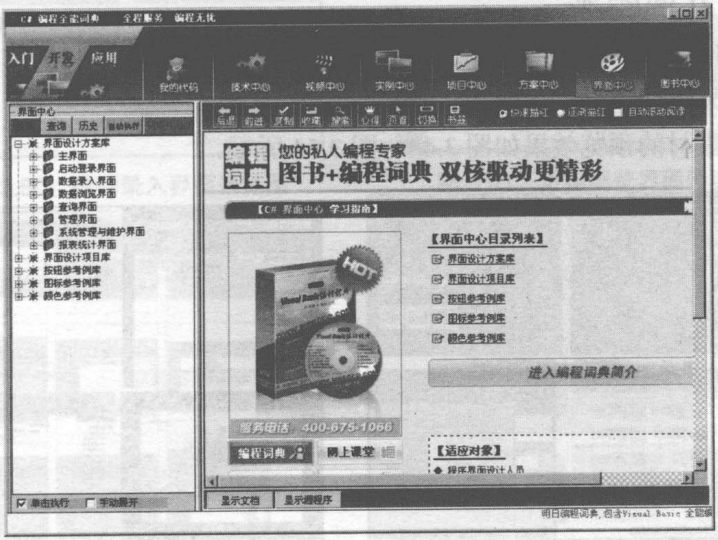


图 3.42　“界面中心”界面运行效果

更多资源请访问稀酷客(www.ckook.com)

“界面中心”提供了界面设计方案库、界面设计项目库、按钮参考例库、图标参考例库和颜色参考例库 5 部分内容。其中包括 **2982** 张界面设计方案、**313** 张图标和按钮、**290** 套项目界面套餐及 **852** 种颜色参考样例等主要开发素材。基本内容结构如图 3.43 所示。

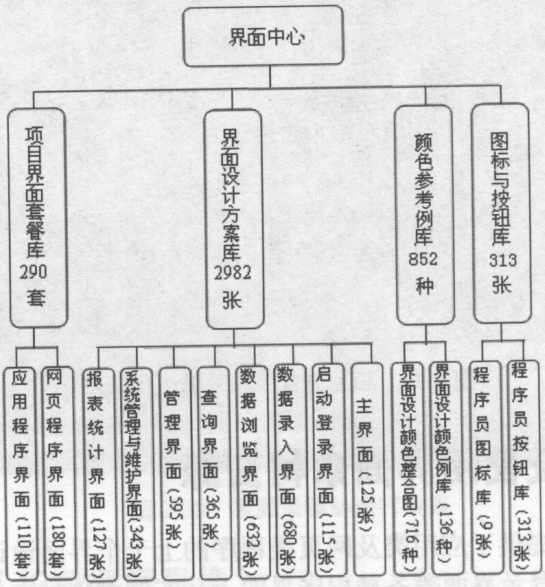


图 3.43 “界面中心”内容结构

3.7.1 界面浏览

以下图片是对所分类界面图片的展示。可以通过界面素材的展示，了解界面中心更多的信息。

部分素材的预览效果如图 3.44～图 3.54 所示。



图 3.44 程序主界面预览图

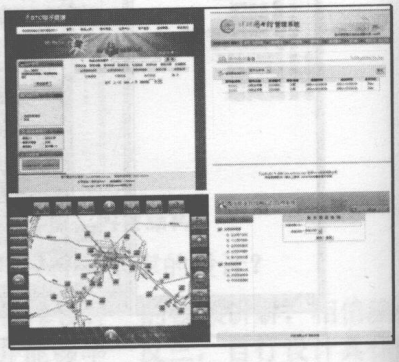


图 3.45 查询界面预览图

更多资源请访问稀酷客(www.ckook.com)



图 3.46 启动、登录界面预览图

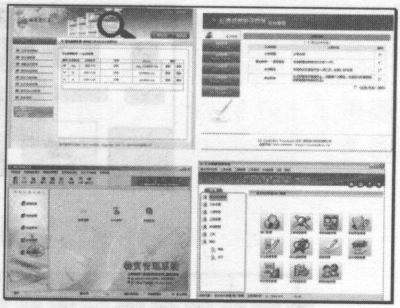


图 3.47 管理界面预览图

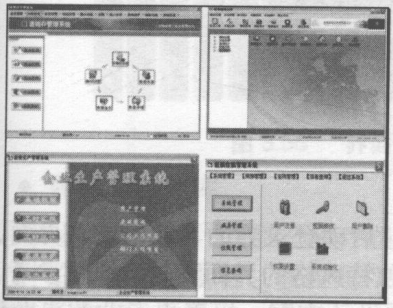


图 3.48 系统管理与维护界面预览图

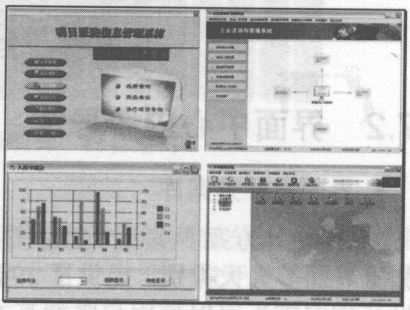


图 3.49 报表统计界面预览图

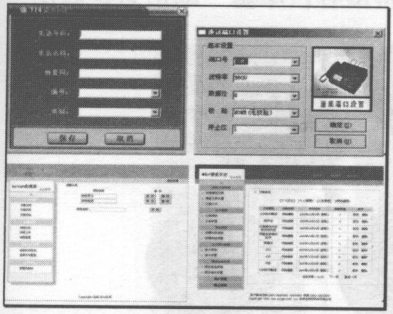


图 3.50 数据录入界面预览图

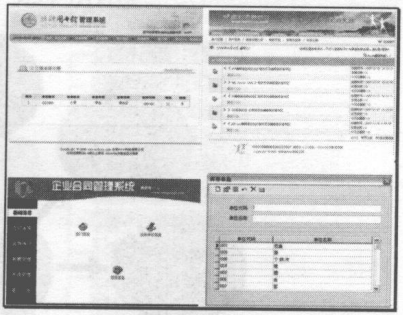


图 3.51 数据浏览界面预览图

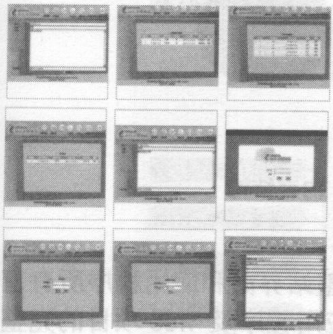


图 3.52 网页界面（邮件发送）



图 3.53 应用程序界面

更多资源请访问稀酷客(www.ckook.com)



图 3.54　图标与按钮预览图

3.7.2　界面中心内容导读

- 界面设计方案库

“界面设计方案”主要分包括主界面、启动登录界面、数据录入界面、数据浏览界面等 8 大类别，提供了 2982 张独特风格的界面素材，通过参考“界面设计方案库”可以使用户掌握基本界面设计方案、培养更专业的界面设计理念。“界面设计方案”基本内容结构如图 3.55 所示。

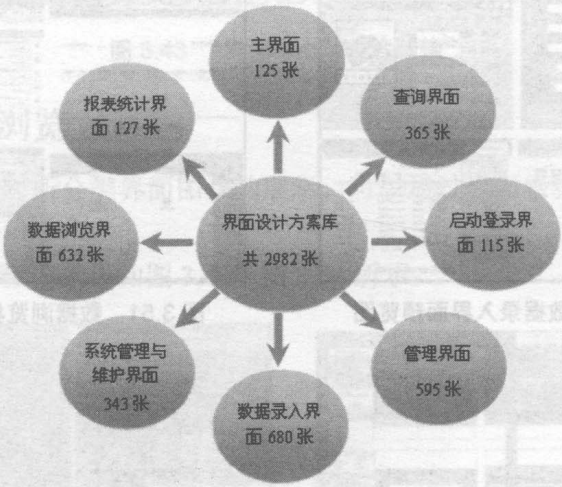


图 3.55　“界面设计方案”内容结构

- 界面设计项目库

“界面设计项目库”主要分为 Web 项目整站界面和应用程序所有窗体界面两大部分。为开发者提供了 280 多套桌面应用程序及 Web 项目的完整界面套餐，为设计和开发人员提供了完备的界面设计解决方案。

更多资源请访问稀酷客(www.ckook.com)

“项目界面套餐库”基本内容结构如图 3.56 所示。

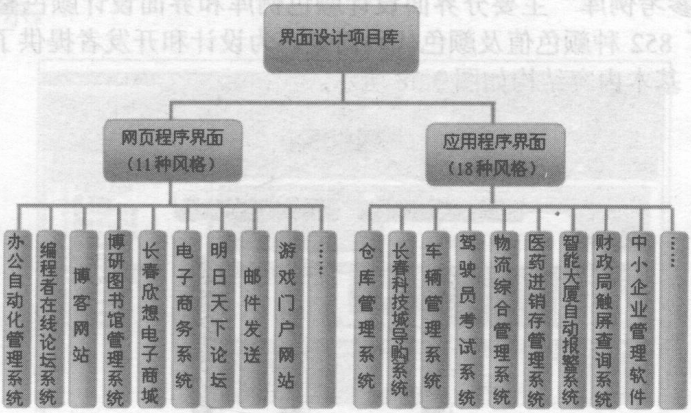


图 3.56 “界面设计项目库”内容结构

- 图标与按钮库

“图标与按钮库”主要分程序员按钮和程序员图标两部分，为开发者提供 313 种图标和按钮素材，覆盖面广、内容丰富，为设计和开发提供了最为优秀的参考方案。

“图标与按钮库”基本内容结构如图 3.57 所示。



图 3.57 “图标与按钮库”的内容结构

更多资源请访问稀酷客(www.ckook.com)

- 颜色参考例库

“颜色参考例库”主要分界面设计颜色例库和界面设计颜色整合图两部分，提供了 852 种颜色值及颜色代码简集。为设计和开发者提供了最优良的配色参考，基本内容结构如图 3.58 所示。

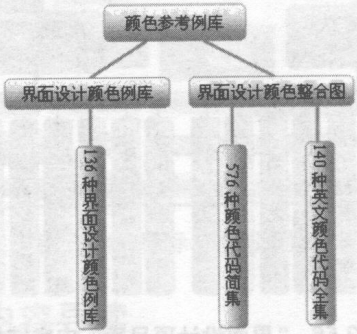


图 3.58 “颜色参考例库”的内容结构

3.7.3 通过目录导航浏览图片

界面中心包括 **8** 种类别的界面。浏览时可通过展开目录，浏览该分类中包含的所有界面的名称。在界面名称目录中找到您想要浏览的界面后直接单击，即可打开相应的界面图片。

操作流程如图 3.59 所示。



图 3.59 浏览图片的操作流程

例如：通过目录导航浏览“启动登录界面”中的图片。

首先展开左侧导航栏中的“界面设计方案库”节点，单击“启动登录界面”展开按钮，即可浏览“启动登录界面”中的目录导航内容。单击左侧导航列表中的界面名称，即可在右侧主窗口中显示该界面的图片信息，如图 3.60 所示。



图 3.60 浏览“启动登录界面”中的图片

更多资源请访问稀酷客(www.ckook.com)

如果选择界面类别名称“主界面”，然后单击导航区上方的“缩略图显示”按钮，右侧主窗体中将会以缩略图的形式显示“主界面”的所有的界面，如图 3.61 所示。



图 3.61　缩略图显示图片



注意

单击缩略图中的某一图片，图片将最大化显示。

如果选择较好的图片，并在图片上单击鼠标右键，在弹出的快捷菜单中选择“图片另存为”命令，可将图片下载，如图 3.62 所示。



图 3.62　保存图片素材

3.8　根据实际需要使用“开发”模式

进入“开发”使用模式以后，用户可以根据自己的实际需求，通过不同

更多资源请访问稀酷客(www.ckook.com)

的方法来使用各版块中的内容。如果您想要查询编程资源，则可以在界面左侧的“查询”区域快速查询；如果您想要学习编程技术，则可以在“技术中心”、“实例中心”、“方案中心”和“项目中心”中通过目录进行学习；如果您想要管理自己的代码，则可以通过“我的代码”版块进行管理。

3.8.1 查询编程资源

用户在编程或学习的过程中经常会遇到某个编程技术掌握得不好，或者程序的某项功能不知该如何实现等情况，这时您可以通过查询功能快速查找编程技术、应用实例、技巧或方案。

在“开发”使用模式的内容导航区中单击“查询”按钮 查询 切换到查询区域，在查询文本框中输入想要查询的关键字，输入查询关键字以后，在内容显示区中将分别显示“参考导航”和“实例导引”区域。

在“参考导航”区域中显示的是与查询关键字相关的编程技术；在“实例导引”区域中显示的则是与查询关键字相关的应用实例。单击“搜索”按钮 [redacted]，或者直接按回车键，在下面的列表框中将显示查询到的结果，用户可以在这三个区域中选择查看所要查询的内容，并且用户还可以对查询的范围、位置及查询结果显示方式等进行相应的设置。

快速查询编程技术的操作流程如图 3.63 所示。

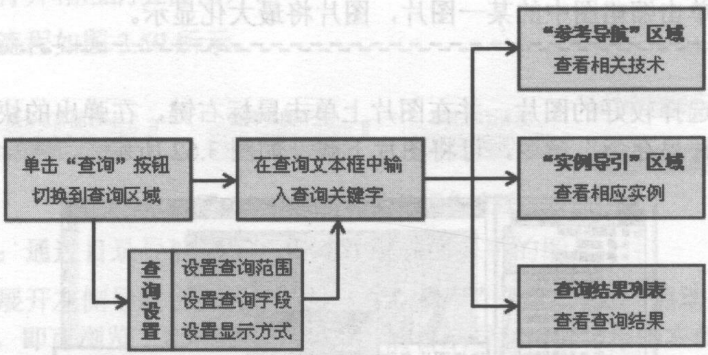


图 3.63 快速查询编程技术操作流程

找到所需要的编程资源以后，通过文档显示区下方的“显示源程序”按钮可以查看源代码，并可以对代码进行复制，然后直接粘贴到 C#开发环境中进行使用，或者进行二次开发。

对于比较重要、需要特别留意的资料，可以通过“收藏”按钮添加到收藏列表，当需要再次查看时，直接通过收藏列表进行快速查看，从而节省了查询的等待时间。通过“心得”按钮，用户还可以记录下学习心得和体会。

更多资源请访问稀酷客(www.ckook.com)

对于界面中心中的界面、图标和按钮等资源，用户也可以另存到本地机器上使用，如图 3.64 所示。

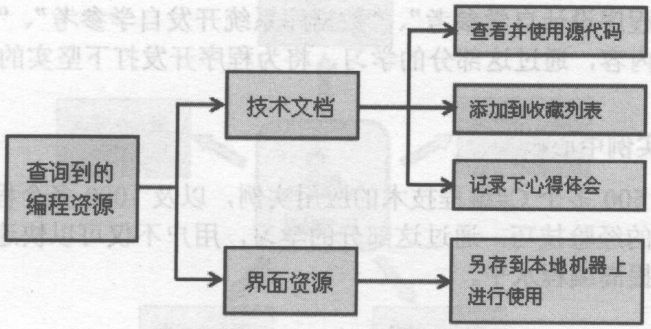


图 3.64　如何使用查询到的编程资源

3.8.2　学习编程技术

如果用户想要学习编程技术，可以在各个中心中通过目录进行学习。在 C#编程全能词典的开发使用模式中提供了学习 C#编程所需要的最全面的编程技术资源库、应用实例资源库、开发项目资源库、应用方案资源库和学习视频资源库。通过这些内容的学习，用户可以逐步掌握 C#编程的各项技术、程序开发的基本方法和经验技巧、项目开发的全过程，从而掌握 C#程序开发的精髓，最终达到能够独立开发项目的目的，如图 3.65 所示。

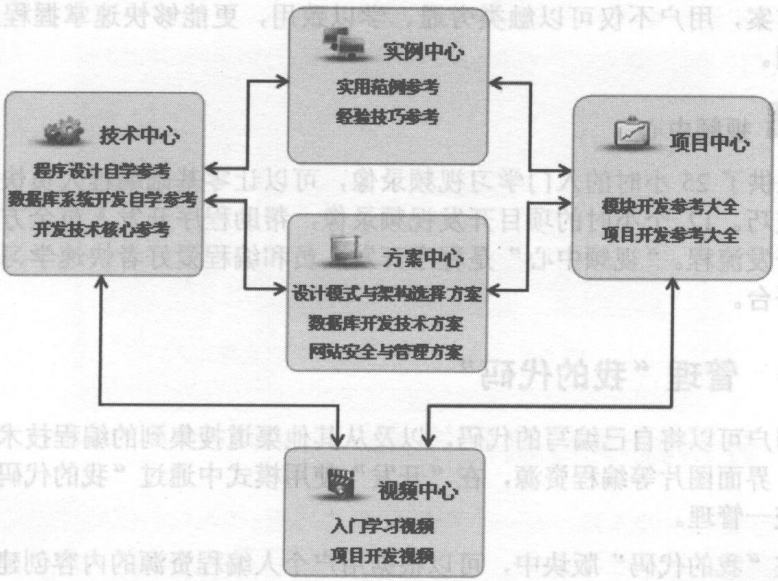


图 3.65　如何学习编程技术

更多资源请访问稀酷客(www.ckook.com)



技术中心

提供了使用C#进行程序开发所需要的最全面、最具权威性的技术文档，内容包括“程序设计自学参考”、“数据库系统开发自学参考”、“开发技术核心参考”等内容，通过这部分的学习，将为程序开发打下坚实的基础。



实例中心

提供了500多个C#编程技术的应用实例，以及1000多个程序开发过程中经常用到的经验技巧，通过这部分的学习，用户不仅可以快速解决开发难题，还可以提高编程水平。



项目中心

通过**25**个行业涉及广泛、实用性强的大型项目案例，详细介绍运用软件工程设计思想开发项目的全过程；在“项目开发基础”和“打包发行参考”中将编程必备的知识与工程开发中的实际案例相结合，使用户在掌握C#基本开发方法和技巧后，能够快速掌握实用项目的开发思路和方法，从而领会C#的精髓。



方案中心

精选了63个实际应用开发方案，包括设计模式与架构选择、页面设计、水晶报表、图形图像化、网络通信、网站安全与管理、网站编译与发布等方案，用户不仅可以触类旁通、学以致用，更能够快速掌握程序开发的精髓。



视频中心

提供了25小时的入门学习视频录像，可以让零基础编程人员快速掌握编程技巧。12个小时的项目开发视频录像，帮助程序开发人员全方位掌握项目开发流程。“视频中心”是程序开发人员和编程爱好者快速学习编程技术的平台。

3.8.3 管理“我的代码”

用户可以将自己编写的代码，以及从其他渠道搜集到的编程技术、应用技巧、界面图片等编程资源，在“开发”使用模式中通过“我的代码”版块进行统一管理。

在“我的代码”版块中，可以根据用户个人编程资源的内容创建不同的分类，进行分类存储，并可以对资源进行查看、添加、修改和删除等操作。

更多资源请访问稀酷客(www.ckook.com)

“我的代码”版块的功能如图 3.66 所示。

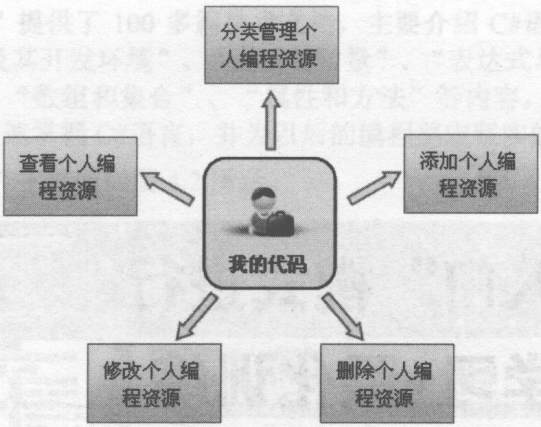


图 3.66 “我的代码”版块的功能

提示

“我的代码”版块的具体使用方法，请参见 2.3.1 节。

更多资源请访问稀酷客(www.ckook.com)

第 4 章

使用“入门”模式进行阶段式学习+强化训练

编程初学者在学习 C#时，需要先从基础学起，然后再学习 C#的核心技术，C#的高级应用，最后学习开发一个完整项目。

C#编程全能词典提供了“入门”使用模式，内容包括 **300** 多篇技术文档，**200** 多套练习题，分入门、进阶、中级和高级 4 个阶段进行学习，通过学习→训练→学习→训练的方式来强化学习，加深对知识的理解和消化，以便轻松掌握实际开发中所需的各种技术，快速提高开发技能。“入门”模式适合于初学编程的爱好者和大中专院校学生使用。

“入门”学习模式的流程图如图 4.1 所示。

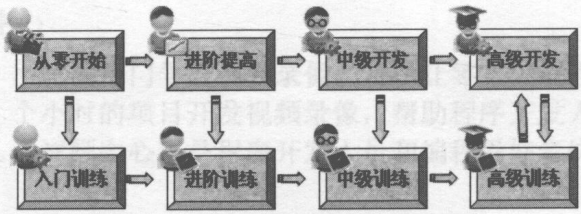


图 4.1 “入门”使用模式学习流程图

4.1 分阶段学习编程技术

学习需要由浅入深，有一个循序渐进的过程。“入门”使用模式的学习部分分为从零开始、进阶提高、中级开发和高级开发 4 个阶段。内容从最基础的语言介绍学起，到核心技术、高级应用，最后到一个完整的项目开发，讲解过程结合注意、说明、技巧、例子等栏目，用户可以轻松掌握，快速入门。

更多资源请访问稀酷客(www.ckook.com)

4.1.1 从零开始

“从零开始”提供了 100 多篇技术文档，主要介绍 C#语言的基础知识，包括“初识 C#及其开发环境”、“变量与常量”、“表达式与运算符”、“字符与字符串”、“数组和集合”、“属性和方法”等内容，通过这部分的学习，用户可以快速掌握 C#语言，并为以后的编程奠定坚实的基础。

“从零开始”主界面如图 4.2 所示。



图 4.2 “从零开始”主界面

默认情况下，单击“从零开始”图标按钮以后左侧目录会自动展开，用户可以通过目录选择要浏览的文档标题，在右侧的内容显示区中浏览文档内容。效果如图 4.3 所示。

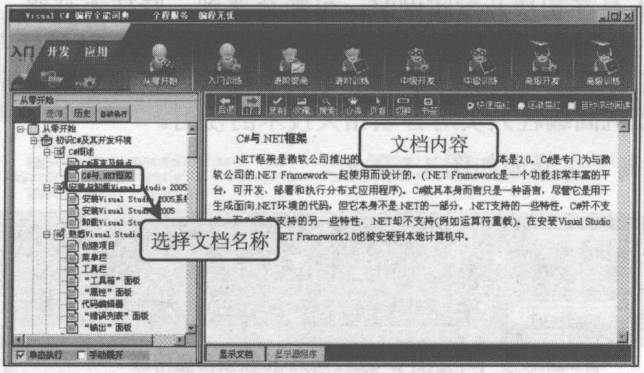


图 4.3 浏览技术文档内容



提示

如果当前文档篇幅过长，读者可以选择文档上方工具栏中的“自动滚动阅读”复选框，文档将实现自动滚动。

更多资源请访问稀酷客(www.ckook.com)

如果需要在技术文档中查询相关技术，可以通过文档上方的“搜索”按钮实现。搜索结果如图 4.4 所示。

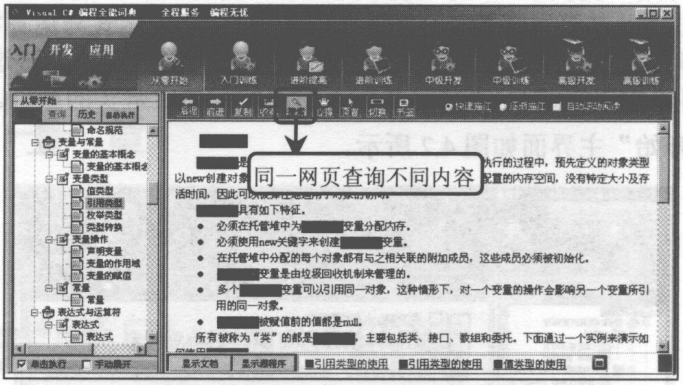


图 4.4　搜索结果界面



注意

如果在同一网页内查询不同技术，将立体描红显示相关查询，即每查询一个技术内容，都采用不同描红颜色描红查询技术，真正达到彻底查询的目的。

4.1.2　进阶提高

C#编程人员在初次开发实例时，可能会遇到一些技术难题，需要参考一些功能类似的示例文档来解决相关问题，“进阶提高”为 C#爱好者提供了 **100** 多篇关于异常处理、Windows 窗体、数据访问技术、程序调试等技术文档，并应用实例来巩固编程人员对各项相关技术的使用。

“进阶提高”主界面如图 4.5 所示。



图 4.5 “进阶提高”主界面

更多资源请访问稀酷客(www.ckook.com)

在讲解核心技术的过程中结合了大量的图示和举例，同时为多数代码提供了注释，帮助用户快速掌握核心技术的具体应用。

例如，选择“进阶提高”/“抽象类与抽象方法”/“抽象类与接口”文档名称，右侧即可浏览关于如何“抽象类与接口”的内容。效果图如图 4.6 所示。

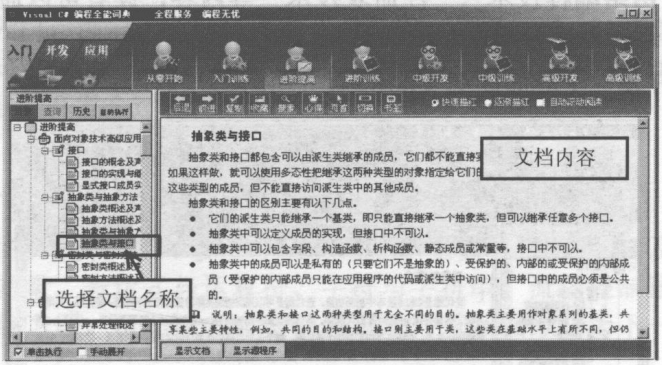


图 4.6　浏览隐藏窗体技术文档



提示

如果用户在关闭程序之前没有浏览完文档，在下次打开程序时还想要继续浏览时，可以为当前文档创建书签。

阅读文档的过程中可以通过文档上方工具栏中提供的“搜索”按钮搜索相关技术，查询结果中关键字以描红的形式进行显示。

例如单击“搜索”按钮，在“搜索”文本框中输入查询关键字“抽象”，查询结果如图 4.7 所示。

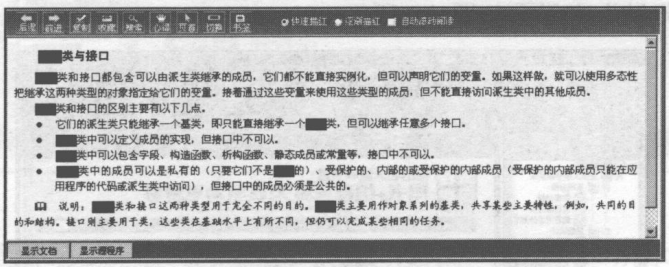


图 4.7　搜索关键字“抽象”



提示

如果文档过长，可以通过文档上方工具栏中的“自动滚动阅读”按钮，以自动滚动的方式浏览文档。

更多资源请访问稀酷客(www.ckook.com)

4.1.3 中级开发

“中级开发”版块提供了100多篇技术文档，主要介绍了“迭代器和分部类”、“泛型的使用”、“文件及数据流技术”、“帮助文件的使用”、“水晶报表与打印”、“网络编程技术”、“注册表技术”等内容，读者可以通过上述内容对C#2.0这些新概念有一个全新的认识。

“中级开发”主界面如图4.8所示。

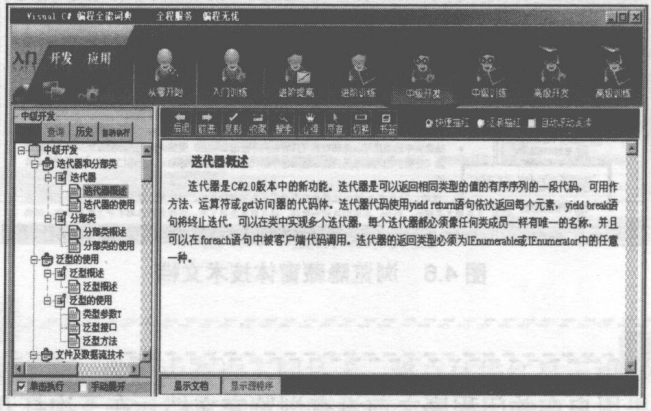


图4.8 “中级开发”主界面

4.1.4 高级开发

学习编程最终的目的就是能够独立开发项目，高级开发提供了**50**多篇技术文档，内容涉及一套完整的项目开发流程。运用软件工程的设计思想，让您学习如何进行软件项目的实践开发，带您一步一步亲身体验开发项目的全过程。

“高级开发”主界面如图4.9所示。



图4.9 “高级开发”主界面

更多资源请访问稀酷客(www.ckook.com)

4.2 通过训练提高开发技能

训练是为了帮助读者更好地巩固和掌握所学的知识，快速提高开发技能，最终能够举一反三，独立开发项目。

训练分入门、进阶、中级和高级 4 个阶段，共提供了 **200** 多套练习题，每套习题以问题+答案的形式进行讲解；同时提供了 **5** 套实用项目练习，介绍不同类型项目的开发全过程。

4.2.1 入门训练

“入门训练”提供了 **80 多**套基础训练题，主要包括初识 C#及其开发环境、变量与常量、表达式与运算符、字符与字符串、流程控制语句、数组与集合等训练习题。帮助读者对基础知识进行重新回顾，为下一步学习打好基础。

“入门训练”主界面如图 4.10 所示。



图 4.10 “入门训练”主界面

每套习题都是以问题+答案的形式进行讲解的。如图 4.11 所示。

提示

为方便读者阅读，文档上方的快捷工具栏中提供了“切换”按钮。它可以实现隐藏目录，大屏幕浏览文档。

更多资源请访问稀酷客(www.ckook.com)

文档浏览以后如果读者需要复制部分代码到开发环境中使用，以加快二次开发的过程，那么读者就可以通过文档下方的“ 显示源程序 ”按钮实现此功能。

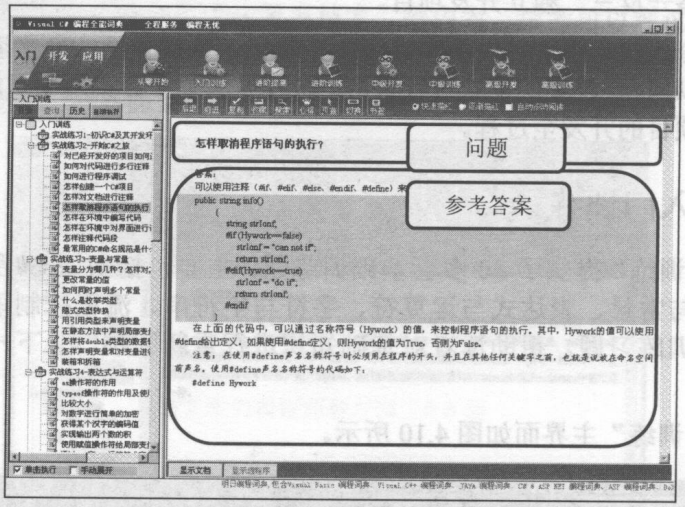


图 4.11　训练题提供方式

“显示源程序”界面的效果如图 4.12 所示。

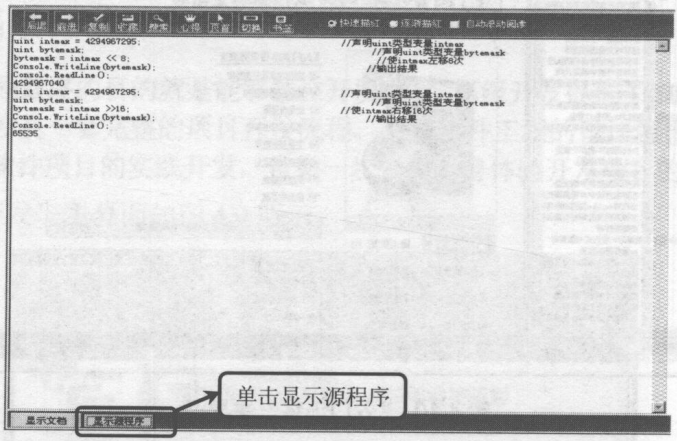


图 4.12　查看程序源代码

在此界面读者通过“复制”按钮或者单击鼠标右键，在弹出的快捷菜单中选择“复制”直接复制，然后在 C#环境中可以直接使用进行二次开发，如图 4.13 所示。

更多资源请访问稀酷客(www.ckook.com)

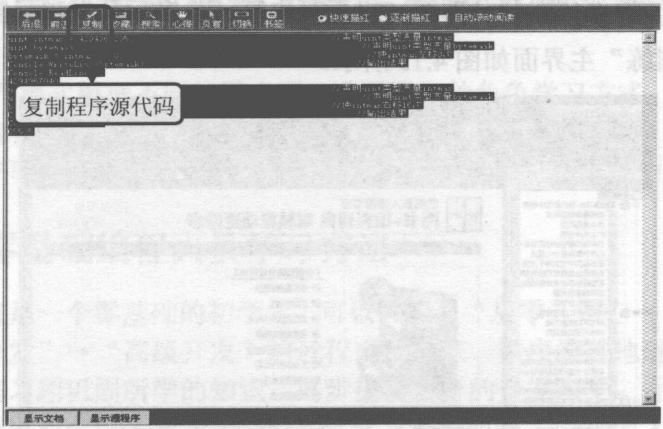


图 4.13　复制程序源代码

4.2.2　进阶训练

“进阶训练”版块提供了 100 套练习题，内容涉及异常处理、Windows 窗体、数据访问技术、Windows 应用程序常用控件、程序调试等，主要用于加深对 C#核心技术的实际应用。

“进阶训练”主界面如图 4.14 所示。



图 4.14　“进阶训练”主界面

4.2.3　中级训练

“中级训练”提供了 100 多套练习题，内容涉及迭代器和分部类、泛型的使用、文件及数据流作用、帮助文件的使用、水晶报表与打印等，可以帮助

更多资源请访问稀酷客(www.ckook.com)

用户更好地了解迭代器和分部类、报表等内容，为以后开发项目打下基础。

“中级训练”主界面如图 4.15 所示。



图 4.15 “中级训练”主界面

4.2.4 高级训练

“高级训练”提供了 **5** 套实用项目练习，内容包括餐饮管理系统、超市进销存销售管理系统、家庭视频监控系统、企业 QQ 系统和在线考试系统 5 个系统的开发全过程，可以帮助读者了解不同类型项目的开发流程，最终实现独立开发项目。

“高级训练”主界面如图 4.16 所示。

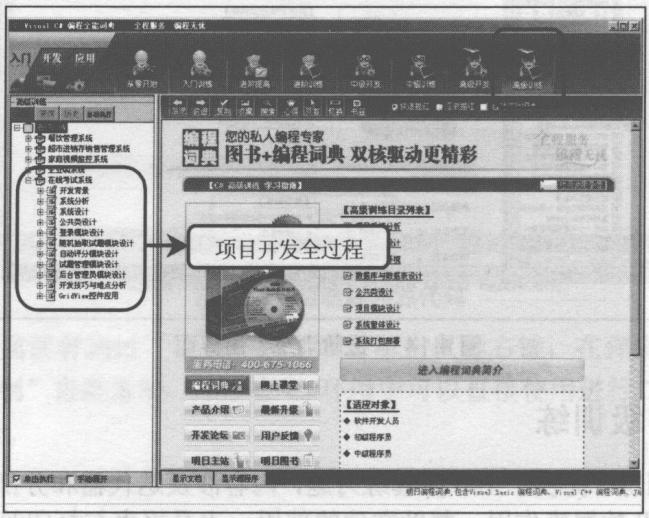


图 4.16 “高级训练”主界面

更多资源请访问稀酷客(www.ckook.com)

4.3 不同水平的编程人员如何使用“入门”模式

“入门”模式根据不同使用人群，设置了三种角色学习方式，用户可以根据自身的编程基础，选择适合自身条件的角色来学习编程知识。通过角色的学习，用户可以及时快速地进行学习和训练，不再为从何学起而发愁。

4.3.1 零基础编程人员学习方法

如果您是一个零基础的初学者，可以按照从“从零开始”→“进阶提高”→“中级开发”→“高级开发”的过程由浅入深、循序渐进地进行学习，然后通过训练习题巩固所学的知识，逐步提高自身的开发水平。并且，每道训练习题都给出相应的参考答案。零基础学习流程如图 4.17 所示。

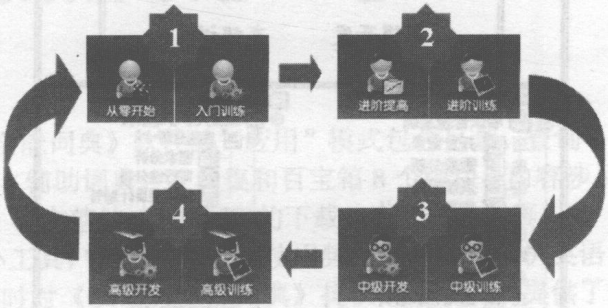


图 4.17 零基础学习流程

4.3.2 中级开发人员使用方法

如果您已经具备一定编程基础，那么可以按照“进阶提高”→“中级开发”→“高级开发”的过程进一步学习 C#编程技术，然后通过训练习题检验自身的编程知识掌握程度，并且能够轻松解决在编程过程中遇到的各种难题，为编写项目打下良好的基础。中级开发者学习流程如图 4.18 所示。

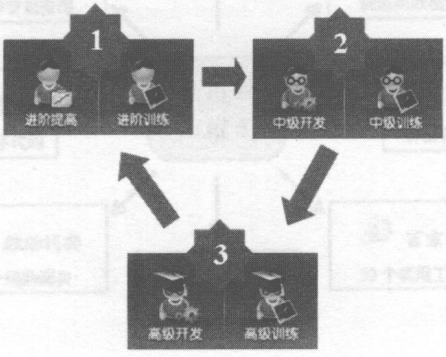


图 4.18 中级开发学习流程

更多资源请访问稀酷客(www.ckook.com)

4.3.3 高级开发人员使用方法

一个高级开发者，在项目开发的过程中经常遇到开发思路和开发流程不够清晰的情况，为了提高项目开发能力，可以使用“入门模式”中提供的“高级开发”版块，该版块提供了一个完整的项目，并清楚地介绍开发项目思路及流程，可以举一反三学会实际开发过程中的各类知识，轻松地完成了训练习题，并通过练习更快地掌握开发各行业项目时，需要提高的个别技术的应用技巧。高级开发者学习内容结构如图 4.19 所示。



图 4.19 高级开发者学习内容结构

更多资源请访问稀酷客(www.ckook.com)

第 5 章

如何使用“应用”模式实现高效开发

《C#编程全能词典》软件的“应用”模式包含目录、查询、检索、源码管理、英语词典、辅助词典、工具集和百宝箱 8 个版块。内容涉及 15 套技术文档的目录、10 套完整的 C#源程序的下载、7 个辅助词典、20 多种工具集和 50 多种实用小工具，收录了编程时使用频率最高的计算机英语及专业英语的汉语解释，同时对《C#编程全能词典》提供的技术文档提供了多关键字的高效查询与智能检索。帮助开发者解决开发难题，快速提高开发效率。

“应用”模式提供的 8 个版块如图 5.1 所示。

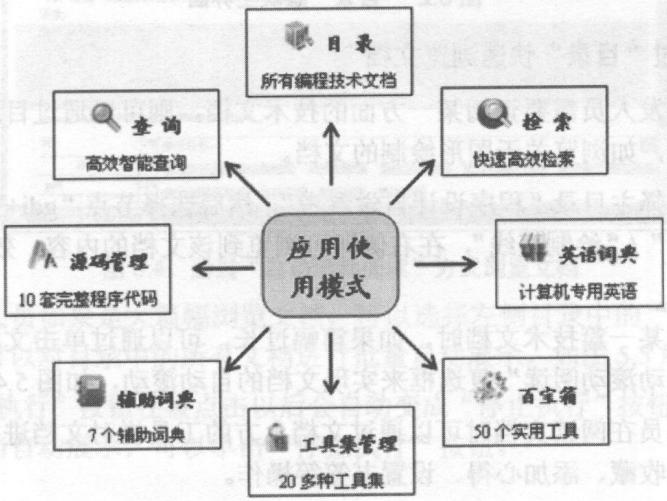


图 5.1 “应用”模式提供的 8 个版块

更多资源请访问稀酷客(www.ckook.com)

5.1 利用目录快速学习和检索编程技术

《C#编程全能词典》为方便编程人员快速学习编程技术，提供了“目录”版块。该版块囊括了**15**套技术文档的目录，其中包括**10000多**条技术文档；**1500多**个模块、实例、技巧；**25**个实用项目。

“目录”版块的主界面如图 5.2 所示。



图 5.2 “目录”版块主界面

1. 通过“目录”快速浏览文档

如果开发人员需要查阅某一方面的技术文档，则可以通过目录进行有选择性的阅读，如浏览关于图形绘制的文档。

首先选择主目录“程序设计自学参考”，然后选择节点“gdi+技术”/“基本图形绘制”/“绘制直线”，在右侧即可浏览到该文档的内容，效果如图 5.3 所示。

在阅读某一篇技术文档时，如果篇幅过长，可以通过单击文档上方工具栏中的“自动滚动阅读”复选框来实现文档的自动滚动，如图 5.4 所示。

开发人员在阅读文档时可以通过文档上方的工具栏对文档进行简单的搜索、文档的收藏、添加心得、设置书签等操作。

更多资源请访问稀酷客(www.ckook.com)

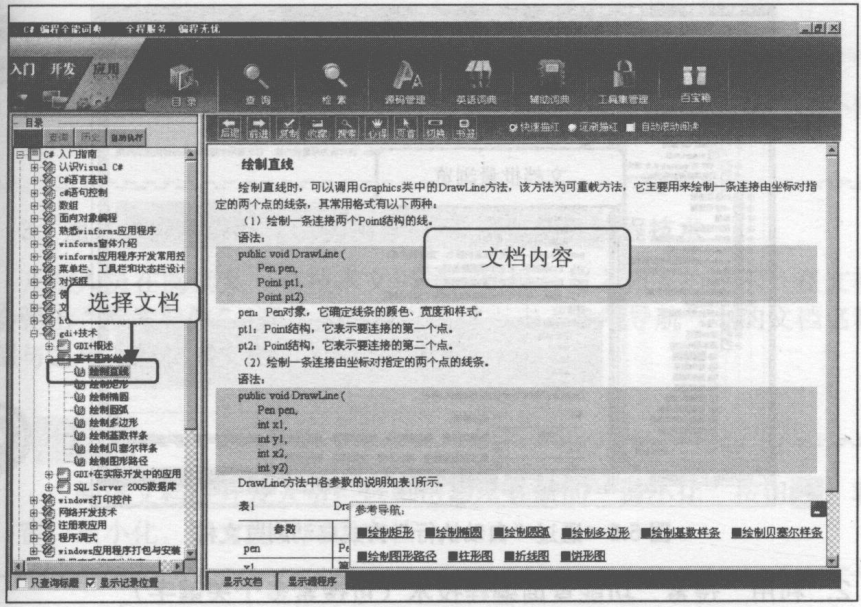


图 5.3　浏览文档内容

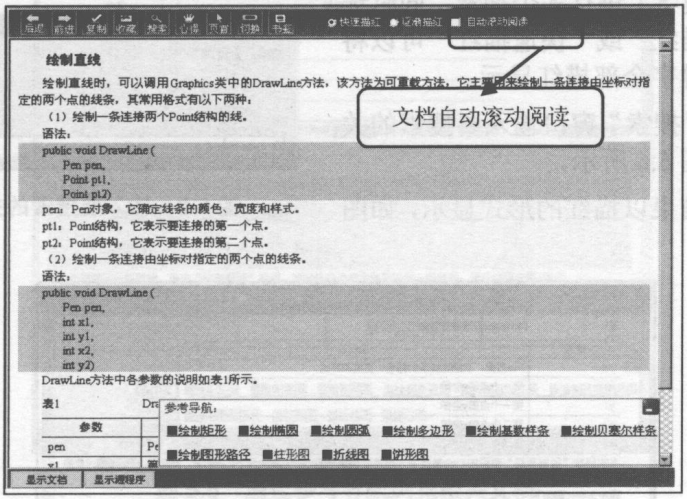


图 5.4　通过“自动滚动阅读”方式浏览文档

开发人员如果是大篇幅浏览文档，可以选择左侧目录中的“自动执行”按钮，它可以对目录中的所有文档进行批量自动展示，如图 5.5 所示。

“自动执行”按钮在被点击以后会自动变成“停止执行”按钮，如果此时不需要文档自动展示，可以单击“停止执行”按钮。

更多资源请访问稀酷客(www.ckook.com)

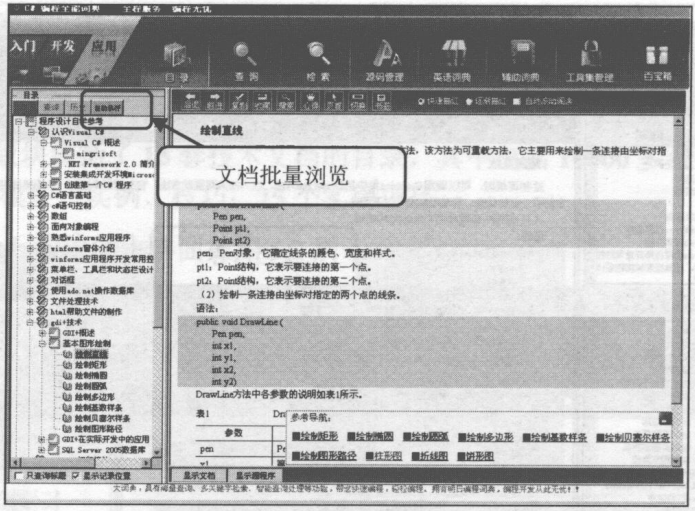


图 5.5　通过“自动执行”方式自动浏览文档

2. 利用“搜索”功能查询编程技术（可搜索多个关键字）

在阅读文档过程中，可以通过文档上方的“搜索”按钮对文档中的关键技术（关键字）进行多个搜索，同时选择“逐渐描红”或“快速描红”可以将查询的关键字全部描红显示。

通过“搜索”窗口输入要搜索的关键字，如图 5.6 所示。

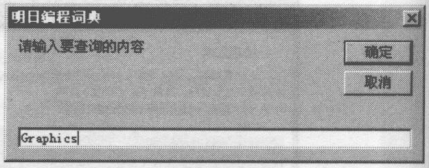


图 5.6　输入要搜索的关键字

查询结果以描红的形式显示，如图 5.7 所示。

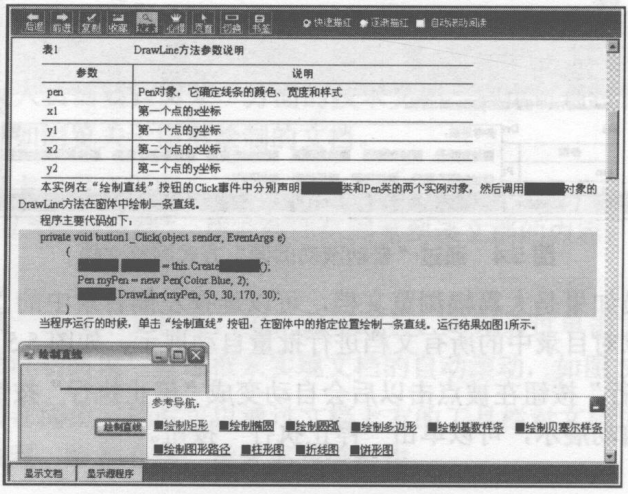


图 5.7　查询结果描红显示

更多资源请访问稀酷客(www.ckook.com)



提示

可以多次单击“搜索”按钮查找多个关键字，查询结果会以不同颜色显示。

3. 通过“参考导航”（“实例导引”）浏览相关编程技术

开发人员在阅读某一篇技术文档时，如果有与之相关的技术，在文档下方会弹出“参考导航”，开发人员可以通过点击“参考导航”中的文档名称来查阅与之相关的技术文档，如图 5.8 所示。



提示

阅读文档时开发人员可以单击参考导航的“最小化”按钮，将窗口最小化，以方便其阅读文档。

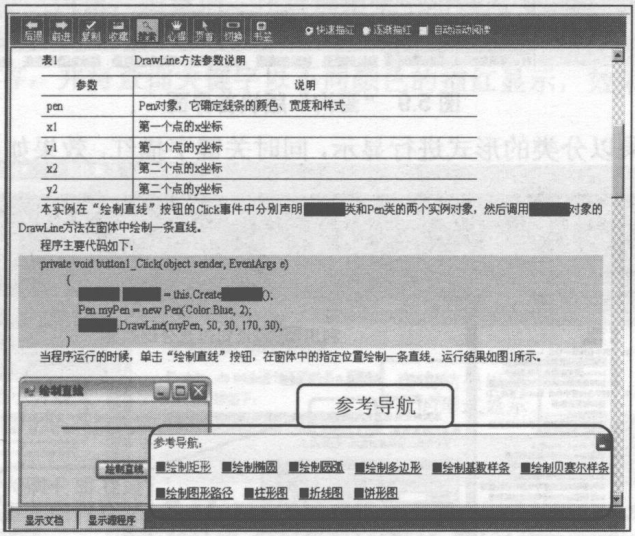


图 5.8　通过参考导航浏览相关文档

5.2　高效查询与智能检索编程技术

5.2.1　通过多种方式查询编程技术

为方便开发者快速高效查询，“查询”版块实现了多种方式的高效查询。其中包括“数据智能查询”、“自动批量数据检索”和“自动提取关键数据”。

更多资源请访问稀酷客(www.ckook.com)

在进行多关键字查询的同时，可以设置“查询范围”和“查询内容”将查询结果准确定位。查询界面如图 5.9 所示。



图 5.9 “查询”版块首界面

查询结果以分类的形式进行显示，同时关键字描红，效果如图 5.10 所示。

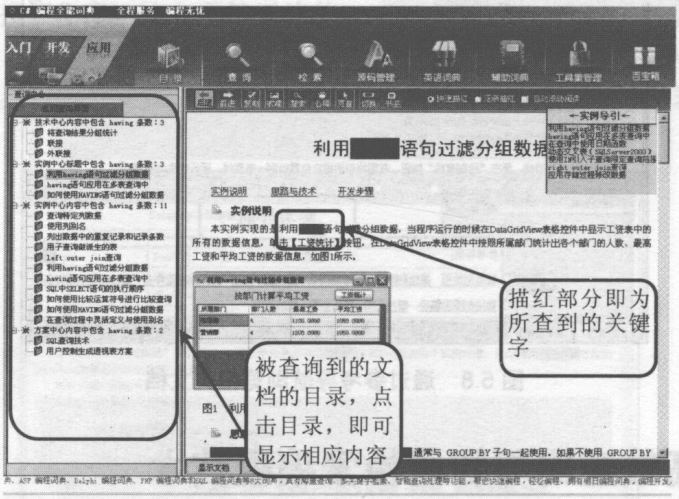


图 5.10 查询结果关键字描红显示

1. 多关键字查询

在“查询”版块中可以同时对多个关键字进行查询。在左侧窗格的文本框中分别输入查询关键字，并选择“与”或“或”；在“选择查询范围”区域

更多资源请访问稀酷客(www.ckook.com)

中选择查询的范围，效果如图 5.11 所示。

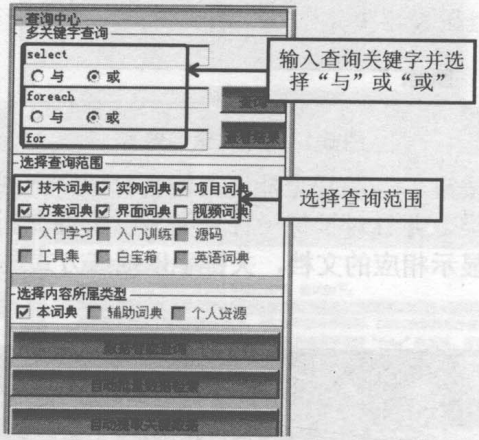


图 5.11　多关键字查询

单击“确定”按钮开始查询。在查询结果中将按照查询范围进行分类，并显示各中心中查询结果的记录条数，单击文档名称在右侧的内容显示区中将显示文档内容，并将查询关键字以不同颜色的描红显示，效果如图 5.12 所示。

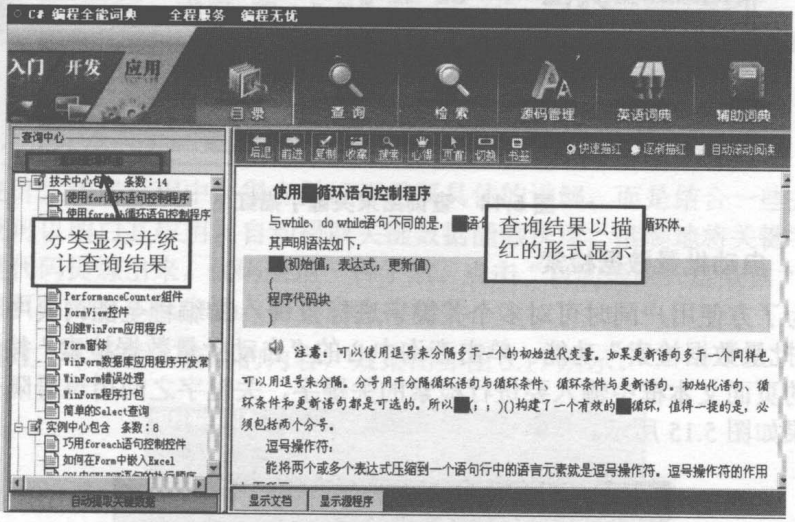


图 5.12　查询结果不同颜色描红显示

2. 数据智能查询

数据智能查询可以将所查到的数据按标题和内容进行分类，方便用户快速应用。单击查询中心中的“数据智能查询”按钮，在弹出页面的文本框中输入要查询的关键字。运行效果如图 5.13 所示。

更多资源请访问稀酷客(www.ckook.com)

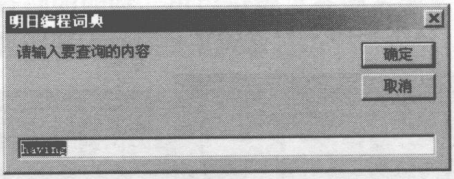


图 5.13　输入查询内容

单击“确定”按钮，C#编程全能词典将会对所查询到的数据按照所属的中心智能地进行分类，并且对于每一种分类显示所查询到的条数。单击左侧的目录，右侧即可显示相应的文档，关键字以描红方式显示，效果如图 5.14 所示。

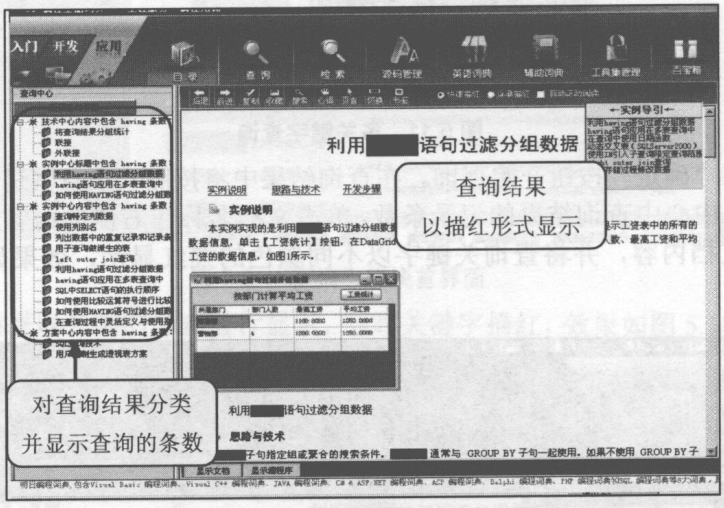


图 5.14　查询结果关键字描红

3．自动批量数据检索

为了方便用户同时可对多个关键字进行查询，C#编程全能词典中设置了“自动批量数据检索”功能。单击查询中心的“自动批量数据检索”按钮，在弹出的页面文本框中输入要进行检索的关键字，关键字之间用逗号隔开，运行效果如图 5.15 所示。



图 5.15　数据智能查询

更多资源请访问稀酷客(www.ckook.com)

单击“确定”按钮进行检索。检索到的文档按照所设置的关键字进行分类，根据查询需要选择相应的关键字，下方目录列出了该关键字的文档名称，单击文档名称，右侧即显示技术文档，其中关键字以描红方式显示。运行效果如图 5.16 所示。

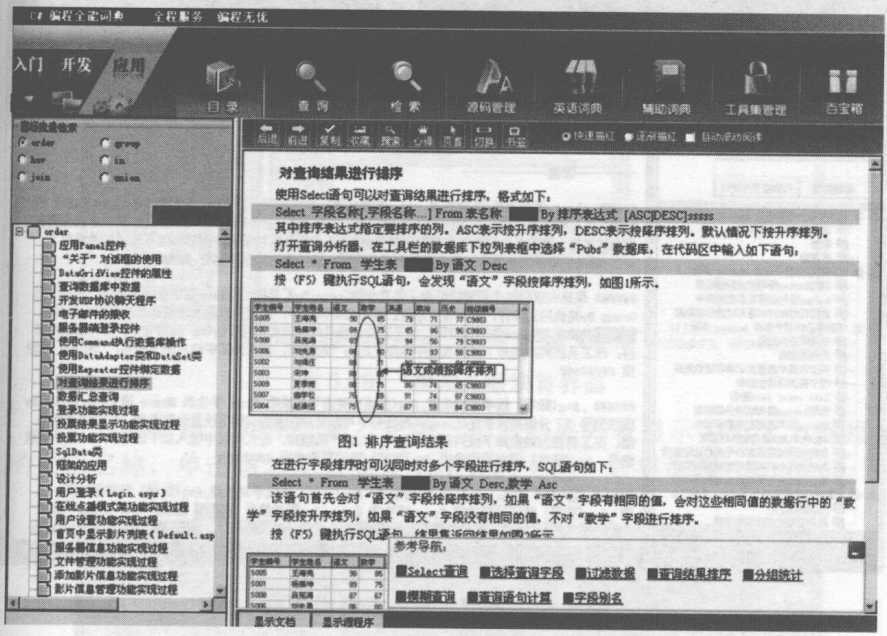


图 5.16　智能查询结果描红显示

4. 自动提取关键数据

在开发程序过程中，很多时候不需要具体的讲解，而是结合一些关键的代码就可以明白其应用，自动提取关键数据能够快速、准确地将关键字相关的关键代码提炼出来，代码应用一目了然。点击“自动提取关键数据”按钮，在弹出页面的文本框中输入要查询的关键字。

通过文本框输入查询的内容，效果图如图 5.17 所示。

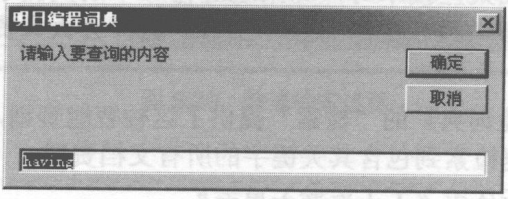


图 5.17　输入要查询的关键字

单击“确定”按钮，C#编程全能词典将自动提取关键字的关键数据。提取数据结束后，右侧显示自动提取的关键数据，用户可以根据关键数据学习

更多资源请访问稀酷客(www.ckook.com)

和应用关键字的使用，关键字以描红方式显示。提取关键数据运行效果如图5.18所示。

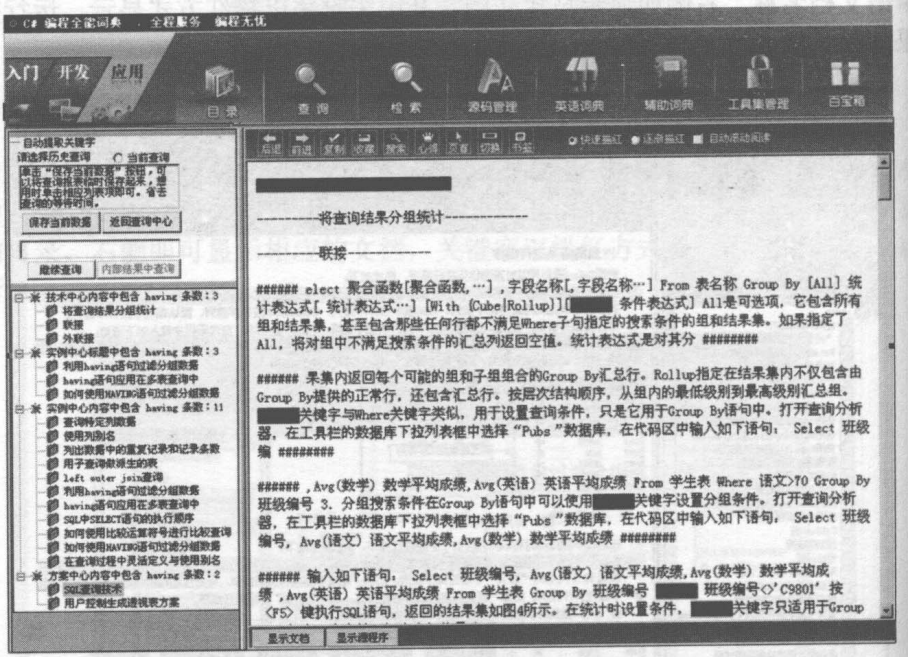


图 5.18 提取关键数据查询结果



技巧

单击“保存当前数据”按钮，可以将查询报表临时保存起来，下次使用时直接单击相应列表项即可，省去查询的等待时间。

5.2.2 通过单个字母（关键字）智能检索编程技术

开发者在查询某些技术时，如果通过输入个别关键字或字母，从而能迅速检索到包含该关键字的所有技术文档，这样会大大提高检索速度及检索的深度。

《C# 编程全能词典》的“检索”提供了这种智能查询，通过输入个别字或字母，即可快速检索到包含其关键字的所有文档资料。

下面介绍如何检索“人力资源主界面”。

如在“搜索”文本框中输入“人”字时，检索结果列表中会列出所有“人”字开头的名称，效果如图5.19所示。

更多资源请访问稀酷客(www.ckook.com)

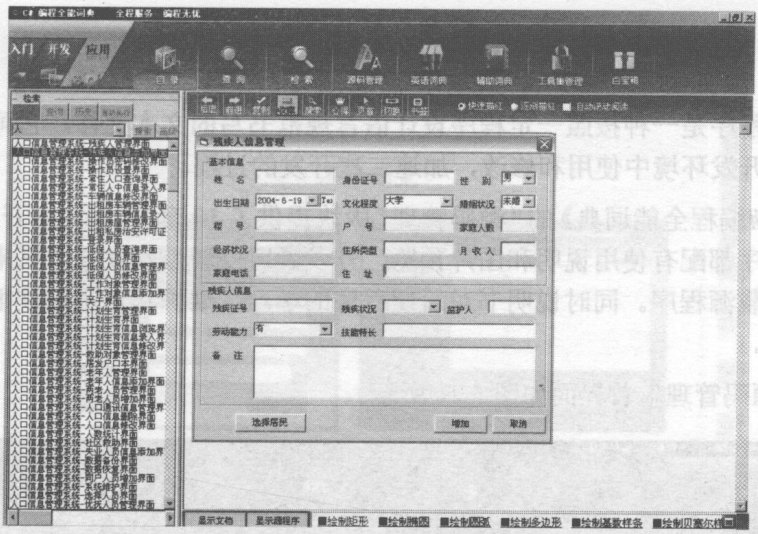


图 5.19 “检索”版块首界面

将“人力资源主界面”输入完整，检索结果列表显示关于人力资源主界面的文档名称，单击该名称右侧显示相关的文档，效果如图 5.20 所示。



图 5.20 快速检索过程

提示

在查询结果中点击“参考导航”中的文档名称还可查看同类别的其他技术文档。

更多资源请访问稀酷客(www.ckook.com)

5.3 有效利用源码实现选择性下载

源程序是一种按照一定程序设计语言规范书写的文本文件，它可以直接复制到开发环境中使用和修改，加速二次开发的过程。

《C#编程全能词典》的“源码管理”版块提供了 **10** 套完整的 C#开发程序，每套程序都配有使用说明和图片预览，开发者可以通过阅读说明书来有选择性地下载源程序。同时说明书对后期程序的学习、维护和使用起到很好的辅助作用。

“源码管理”首界面如图 5.21 所示。



图 5.21 “源码管理”版块首界面

在下载源程序或直接打开开发环境使用之前，可以通过点击文档附带的使用说明书浏览一下程序的各种界面及具体使用说明，以便有选择性地下载或使用。使用说明书的效果如图 5.22 所示。

通过目录选择程序名称，便可以打开程序所在的文件夹，如果用户的计算机中安装了 C#的运行环境，可直接运行程序。打开源程序效果如图 5.23 所示。

对于没有安装开发环境的开发者，可以将其下载至本地机器，在安装了开发环境以后再使用。

更多资源请访问稀酷客(www.ckook.com)

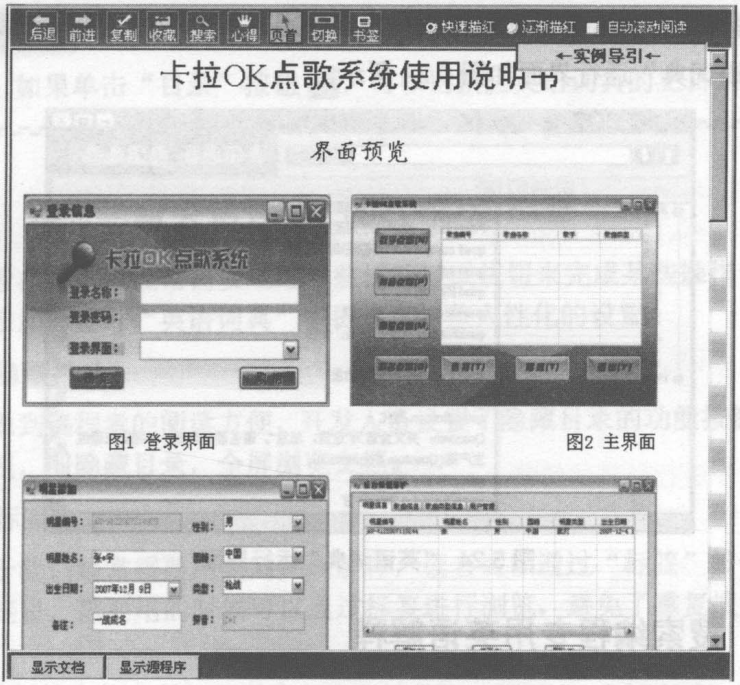


图1 登录界面　　图2 主界面

图 5.22　程序使用说明书

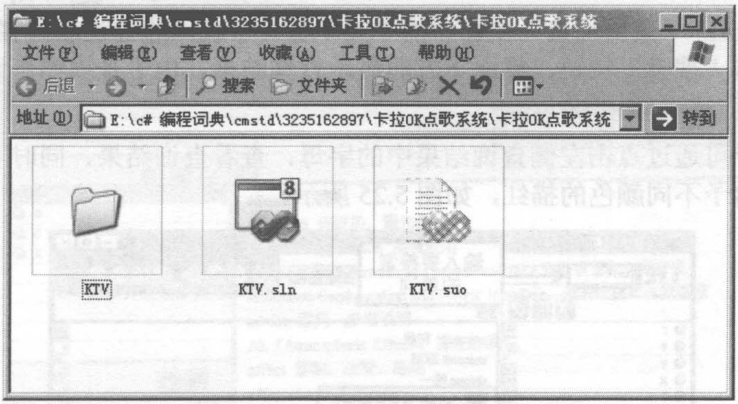


图 5.23　源程序界面

5.4　计算机（专业）英语速查

很多程序员在编程时都会发现，可能由于英语水平比较低，加上编程有专业的英语，计算机有计算机专用英语，这样由于手头的词汇量不够，会给程序的开发带来一定的难度。

“英语词典”收录了编程时使用频率最高的计算机英语及专业英语的汉语

更多资源请访问稀酷客(www.ckook.com)

解释，可对用户搜索的单词进行模糊查询。用户可以随用随查，方便快捷。

“英语词典”运行界面如图 5.24 所示。

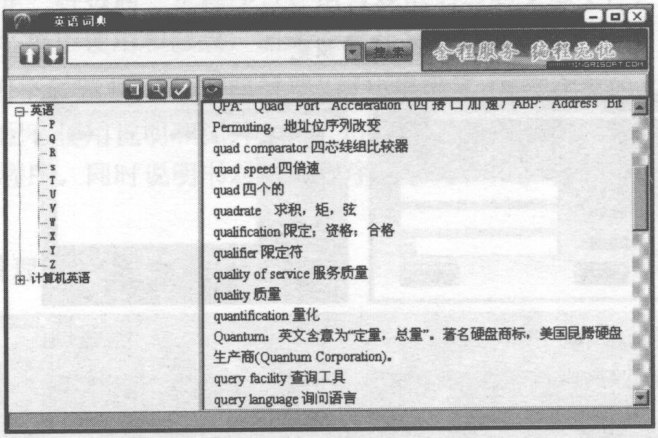


图 5.24 “英语词典”运行界面

5.4.1 搜索编程专用英语解释

在“英语词典”主界面，开发人员可以直接通过输入英文单词搜索其相关解释。如查询“unit”的词解：

首先在 “英语词典”主窗口的“文本框”中输入要查询的内容：unit，然后点击“搜索”按钮，界面下方即显示查询结果。由于是模糊查询，查询结果可能会包含很多字母。

用户可通过点击左侧查询结果中的字母，查看查询结果，同时对于查询关键字给予不同颜色的描红，如图 5.25 所示。

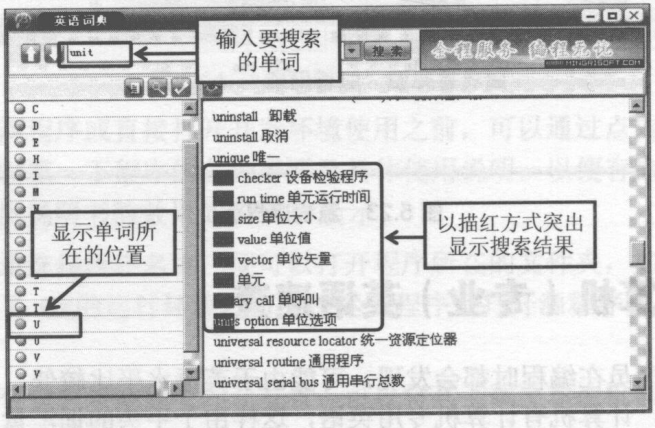


图 5.25 查询结果

更多资源请访问稀酷客(www.ckook.com)



提示

如果单击“目录”按钮，可以切换回英语词典的主目录。

5.4.2 如何隐藏英语词典目录、添加标签

读者在浏览时经常需要通过一些快捷键或按钮来完成某些操作（如阅读方式、添加标签），“英语词典”中提供了一些人性化的设置。

- 隐藏目录

考虑到编程者的阅读方便，开发人员设置了隐藏目录的功能按钮，单击此按钮，即隐藏目录，全屏浏览文档。

- 标签

对于我们经常需要用到的英语解释，读者可以通过“标签”按钮将其添加在标签里，需要用的时候可以通过标签进行浏览，避免了重复搜索带来的麻烦。

如浏览或搜索到比较好的文档时，首先选择单击此文档名称，然后单击“标签”按钮，进入标签界面，在下方的文本框中输入设置的主题名称。最后单击添加按钮即可。效果如图 5.26 所示。

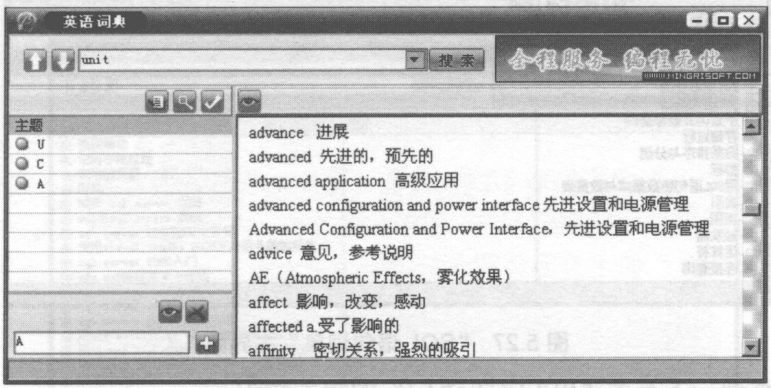


图 5.26 添加标签



注意

如果单击界面下方的删除按钮可以删除已经设置的标签。

更多资源请访问稀酷客(www.ckook.com)

5.5 使用多种辅助词典协助编程

作为一名编程人员，SQL 语句、数据库技术、UML、程序测试和编码规范等知识是编程的必修课。

《C#编程全能词典》提供了 7 种辅助词典，包括 SQL 语句词典、SQL 数据库技术词典、UML 词典、软件工程词典等，这些辅助词典中提供了在编程时所能涉及的技术文档，用户可即用即查。

5.5.1 SQL 语句词典

SQL 是几乎所有的关系型数据库都支持的语言，编程人员可通过 SQL 语句对数据库进行日常的管理（如数据录入、数据查询、子查询与嵌套查询、排序和分组，以及游标、视图、触发器和存储过程的创建及维护）。

“SQL 语句词典”包括了所有数据库相关的 SQL 语句。编程人员可通过“SQL 语句词典”查询与之相关的所有 SQL 语句。

“SQL 语句词典”主界面如图 5.27 所示。

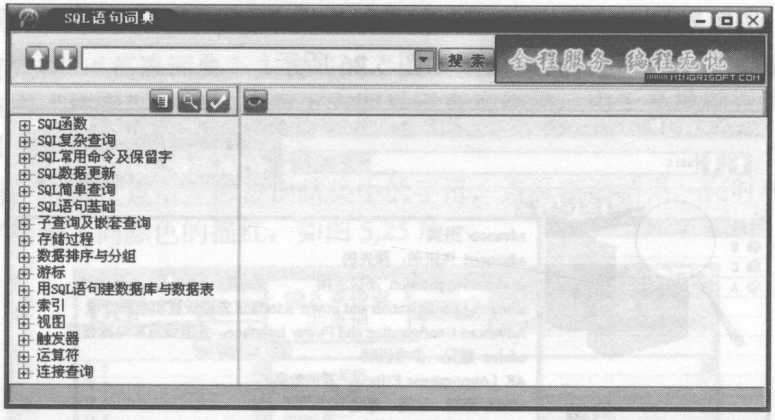


图 5.27 “SQL 语句词典”主界面

如查询一下有关“SELECT 语句”的相关知识。

首先在“SQL 语句词典”主窗口的“文本框”中输入查询内容：SELECT，然后单击“搜索”按钮，界面下方显示查询结果。单击其中的“SELECT 命令”文档名称，右侧即显示文档内容，同时对于查询关键字给予不同颜色的描红。查询结果如图 5.28 所示。

更多资源请访问稀酷客(www.ckook.com)

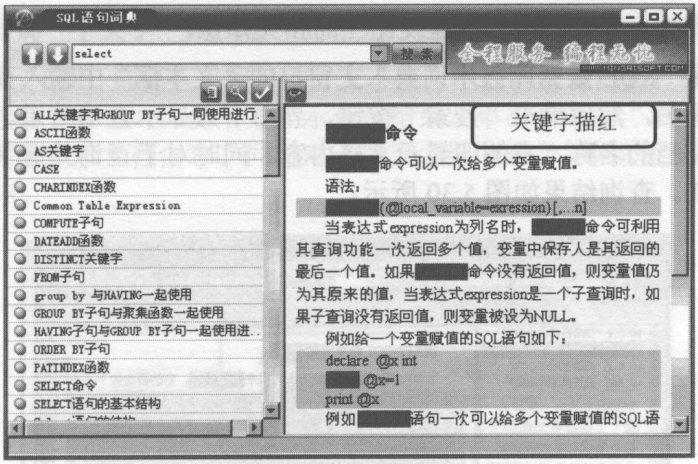


图 5.28　查询结果

5.5.2　SQL 数据库技术词典

在开发程序时，避免不了和数据库打交道，SQL 数据库是常用的一种数据库。视图、游标、存储过程、触发器、数据库的操作和使用等都是数据库的重要内容。

“SQL 数据库技术词典”包括了与数据库相关的所有知识（如视图、游标、存储过程、触发器、创建和使用索引等）。

“SQL 数据库技术词典”的主界面如图 5.29 所示。

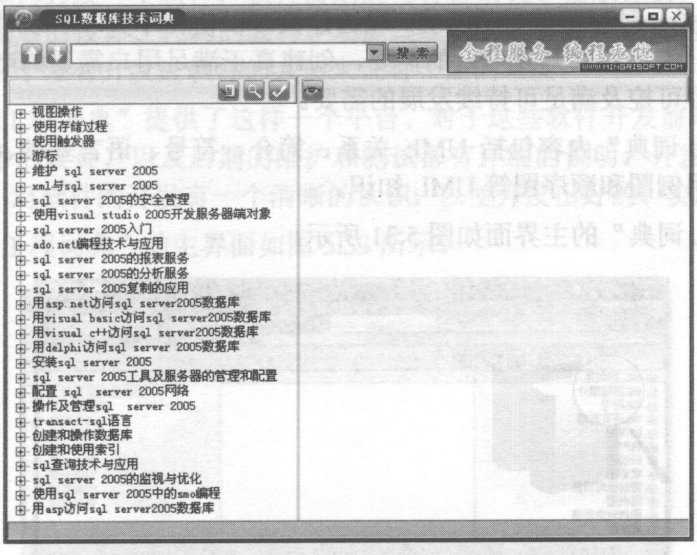


图 5.29 “SQL 数据库技术词典”的主界面

更多资源请访问稀酷客(www.ckook.com)

如查询一下有关如何“创建视图”的相关知识。

首先在“SQL 数据库技术词典”主窗口的“文本框”中输入查询的关键字：创建视图，然后单击“搜索”按钮，界面下方显示查询结果。单击其中的某一个文档的名称，右侧即显示文档内容。同时对于查询关键字给予不同颜色的描红。查询结果如图 5.30 所示。

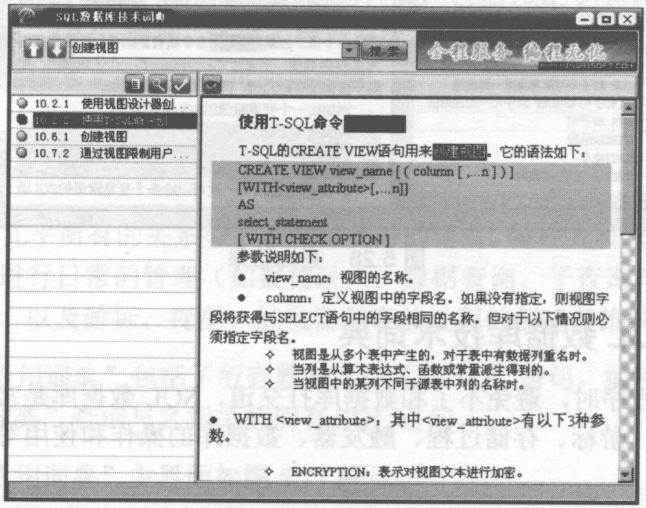


图 5.30　查询结果

5.5.3　UML 词典

UML 是一种标准的建模语言，通过建模可以使程序员理解软件系统的概念和原理、满足协同开发项目的需要，创建真正满足用户需要的产品，同时使开发过程可控及满足可持续发展的需要。

“UML 词典”内容包括 UML 关系、简介、符号、语言基础、协作图、活动图、用例图和顺序图等 UML 知识。

“UML 词典”的主界面如图 5.31 所示。

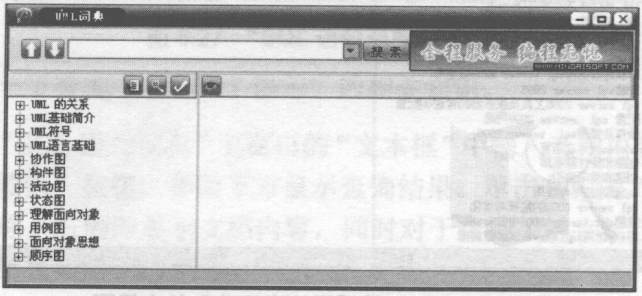


图 5.31 “UML 词典”的主界面

更多资源请访问稀酷客(www.ckook.com)

下面查询一下关于“用例图”的相关内容。

首先在“UML 词典”主窗口的“文本框”中输入查询内容：用例图，然后单击“搜索”按钮，界面下方显示查询结果。单击其中的“用例图”文档名称，右侧即显示文档内容。同时对于查询关键字给予不同颜色的描红，如图 5.32 所示。

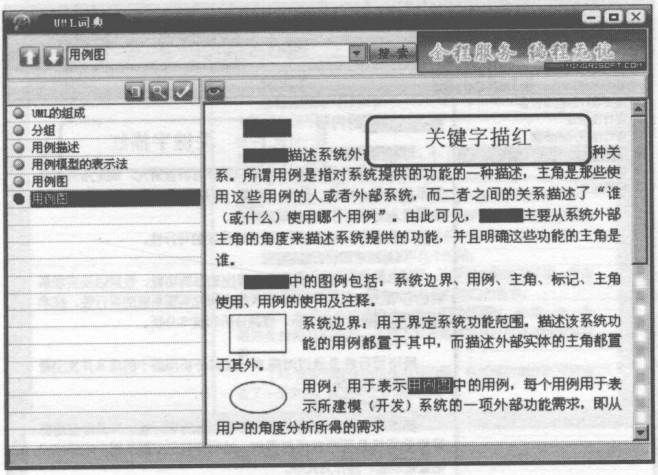


图 5.32　查询结果

5.5.4　软件工程词典

对于如何开发一个软件，软件前期的可行性研究、需求分析；软件开发过程中数据库的设计、代码的编写、功能模块的详细设计；后期的测试及维护，这些对于没有开发过软件的人员来说有一定的难度。

“软件工程词典”提供了这样一个平台。对于这些软件开发前期的分析、开发过程中的设计，以及后期的维护和测试都有详细的说明。开发人员可以通过查询，对软件的开发有一个清晰的认识，以便开发出好的、实用的软件。

“软件工程词典”的主界面如图 5.33 所示。

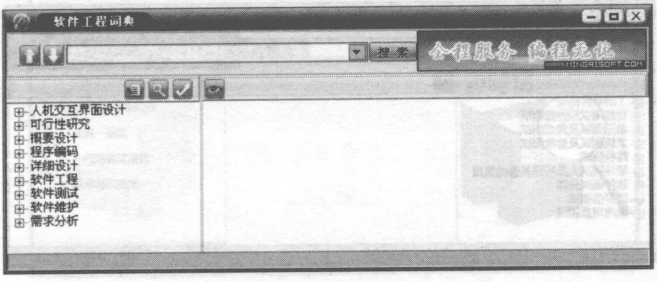


图 5.33　“软件工程词典”的主界面

更多资源请访问稀酷客(www.ckook.com)

下面查询关于如何进行软件前期的可行性研究。

首先在“软件工程词典”主窗口的“文本框”中输入查询的关键字：可行性研究，然后单击“搜索”按钮，界面下方显示查询结果。单击其中的“可行性研究的内容”文档名称，右侧即显示文档内容。同时对于查询关键字给予不同颜色的描红，如图 5.34 所示。

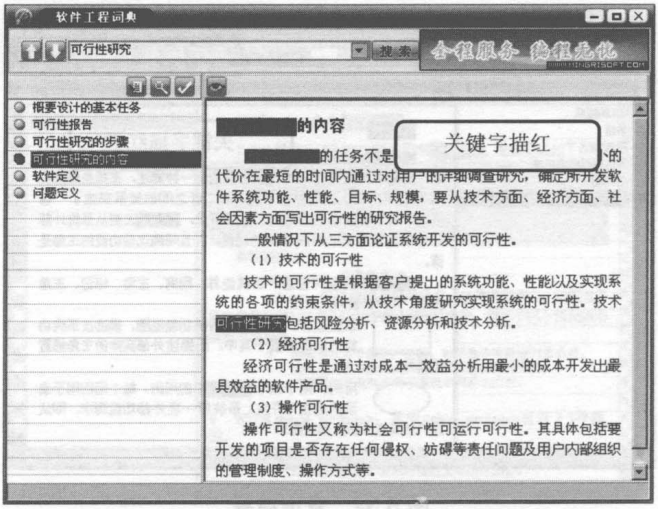


图 5.34　查询结果

5.5.5　程序测试词典

通常程序员在编写完程序以后，后期的测试是一个很重要的工作。软件测试人员测试的内容（如功能测试、性能测试、配置测试和安全测试等）需要分很多类。

在“程序测试词典”中，编程人员可以对测试要求、测试的分类等具有一个明确的认识，以便程序在测试以后能真正做到方便、实用、安全。

“程序测试词典”的主界面如图 5.35 所示。

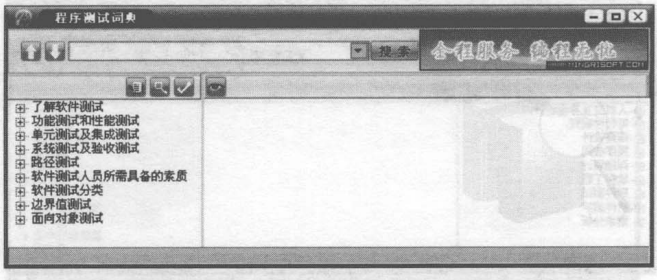


图 5.35　“程序测试词典”的主界面

更多资源请访问稀酷客(www.ckook.com)

下面查询一下关于“安装测试”方面的文档。

首先在“程序测试词典”主窗口的“文本框”中输入查询的关键字：安装测试，然后单击“搜索”按钮，界面下方显示查询结果。单击其中的“安装测试”文档名称，右侧即显示文档内容。同时对于查询关键字给予不同颜色的描红。查询结果如图 5.36 所示。

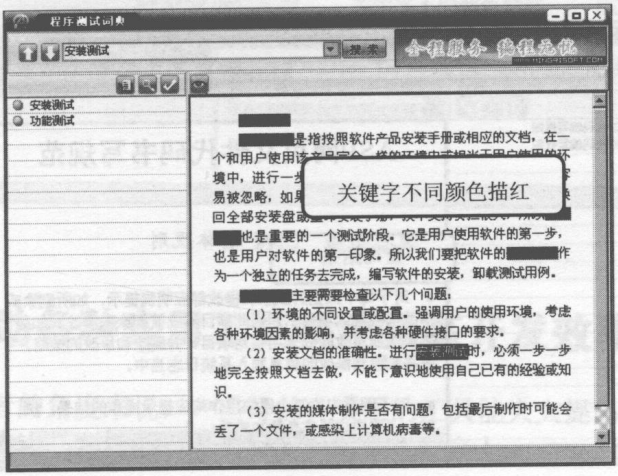


图 5.36　查询结果

5.5.6　编码规范词典

编程人员在编写程序时，代码的编写（如对象命名、代码书写、代码注释）必须遵循一定的规范，而且规范代码的编写其实也是养成良好编程习惯的基础。

《C#编程全能词典》为编程人员提供“编程规范词典”，程序员在编程时可以通过“编程规范词典”对代码编写的一些规则进行查询。

“编码规范词典”的主界面如图 5.37 所示。

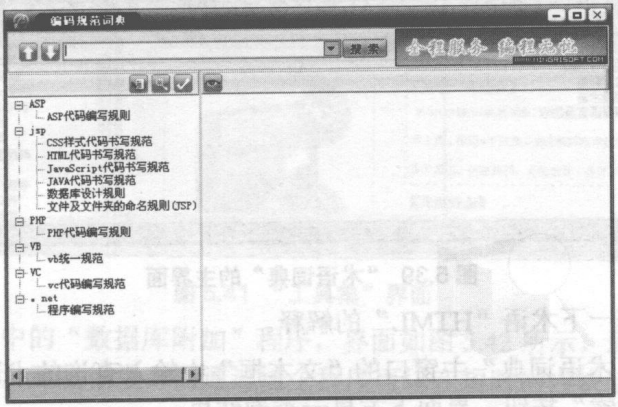


图 5.37 “编码规范词典”的主界面

更多资源请访问稀酷客(www.ckook.com)

下面查询一下代码编写规则。

首先在“编程规范词典”主窗口的“文本框”中输入查询的关键字：代码编写规则，然后单击“搜索”按钮，查询结果中会显示不同语言的编写规则。单击其中的“ASP 代码编写规则”文档名称，右侧即显示文档内容，如图 5.38 所示。

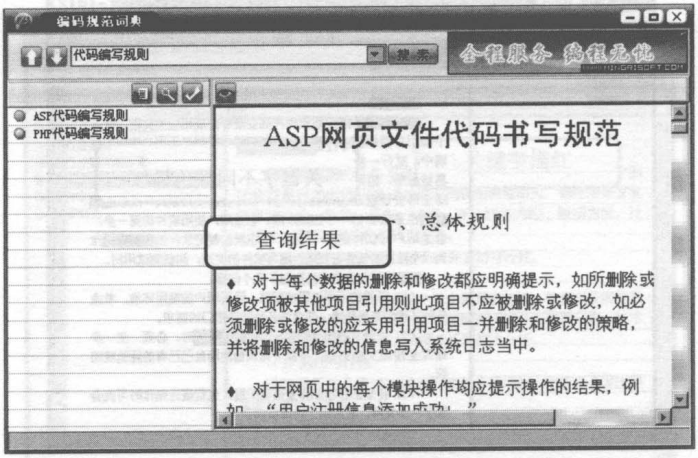


图 5.38　查询结果

5.5.7　术语词典

编写人员在使用各种语言（如 ASP.NET、C#、Visual Basic ）进行编程时可能会碰到各种各样的专业术语，尤其是那些英文缩写让开发人员不知所云。

《C#编程全能词典》为编程者提供了“术语词典”，编程人员可以查询相关语言的术语解释。“术语词典”的主界面，如图 5.39 所示。

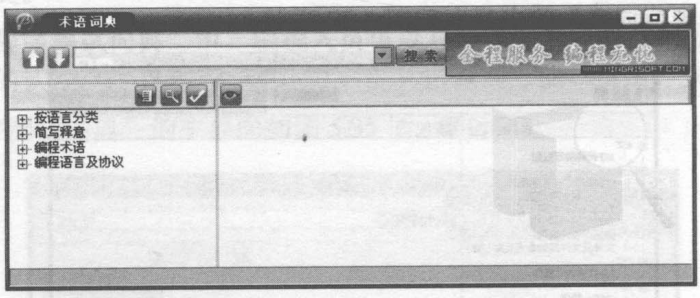


图 5.39 “术语词典”的主界面

下面查询一下术语“HTML”的解释。

首先在“术语词典”主窗口的“文本框”中输入查询的术语：HTML，然后单击“搜索”按钮，界面下方显示查询结果。

更多资源请访问稀酷客(www.ckook.com)

如单击其中的“HTML 语言”文档名称，右侧即显示文档内容。对于关键字给予不同颜色的描红，如图 5.40 所示。

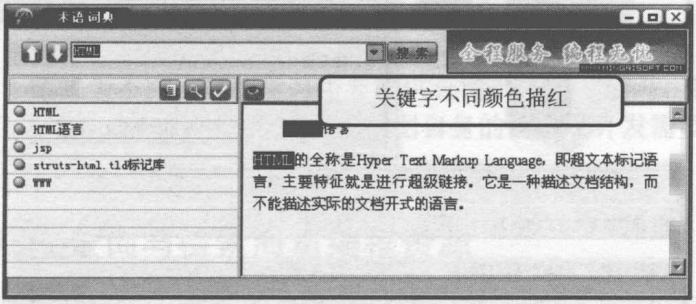


图 5.40　查询结果

5.6　借助各种实用工具快速提高开发效率

从开发者的角度来说，一些好的、实用的工具能大大提高开发者的工作效率。从而能让开发者把更多的时间用在开发程序上。

“工具集”版块向开发者提供了 **21** 个实用工具，包括图片分类存储、图片批量预览、文件拷贝器、备份数据源等数据库类工具。“工具集”界面如图 5.41 所示。



图 5.41 “工具集”界面

如工具集中的“数据库附加”程序，界面如图 5.42 所示。通常开发者手动附加数据库，速度很慢，而通过这个程序可以在短时间内快速完成数据库的附加操作。

更多资源请访问稀酷客(www.ckook.com)



图 5.42　附加数据库

这里开发者只需找到源数据库和目标数据库位置，然后选择数据库文件（*.MDF 和*.LDF），最后选择“开始附加数据库”按钮即可实现数据库附加操作，如图 5.43 所示。



图 5.43　数据库附加界面

5.7　有效通过小工具解决实际问题

在程序的开发过程中，开发者经常需要通过一些小工具来解决某些实际问题，如修改文档名称、修改 IP 地址等。

“百宝箱”版块收录了 50 多个实用小工具，内容涉及图片处理、重命名、数据库等多方面内容。

更多资源请访问稀酷客(www.ckook.com)

5.8 不同人群如何使用“应用”模式进行快速、高效的开发

“应用”模式为不同人群的学习需要提供查询、检索、源码、辅助词典、工具集和百宝箱等 8 个版块。用户可以根据自身的情况及开发需要选择不同的版块进行学习。

5.8.1 高效查询与检索海量编程资源

作为一名有着两三年经验的编程人员，我经常需要查询一些编程的技巧、技术、实例、项目及方案的应用，一是可以节省程序的开发时间，二是在一定程度上会提高我的编程能力。

最近一段时间我在做一个 “图书馆管理系统”的项目，但由于手头资料有限，加上开发时间短，我想在短时间内查找一些相关资料（如技术、实例、源程序、图片等）直接修改并使用。

在《C#编程全能词典》的“应用”模式中我找到了想要的资料。“应用”模式的“查询”版块运行界面如图 5.44 所示。



图 5.44 “查询”版块运行界面

应用模式的“查询”版块提供了 4 种查询：普通查询（多关键字）、数据智能查询、自动批量数据检索、自动提取关键数据。

下面是我通过“数据智能查询”查询到的关于图书馆管理系统的资料。

首先在“应用”模式中单击“数据智能查询”图标按钮，弹出如图 5.45 所示的界面。在文本框中输入“图书馆”字样，单击“确定”按钮进行查询。

更多资源请访问稀酷客(www.ckook.com)

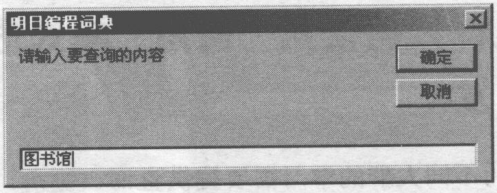


图 5.45　输入查询内容

查询结果出乎我的意料之外，不但查询的速度快，而且查询到的相关资料很多，如图 5.46 所示。查询结果中分类显示了每个中心查询的结果及查询条数，同时还查询到完整项目的开发全过程。

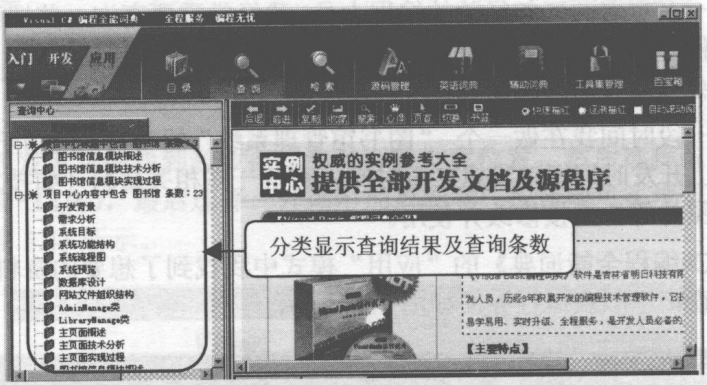


图 5.46　查询结果

有了这些资料，我轻而易举地实现了“图书馆管理系统”程序的开发。

提示

经过查询发现“自动批量数据检索”用于同时对多个关键字进行查询，关键字以描红形式显示；“自动提取关键数据”可以快速、准确地将文档中与关键字相关的文字或关键代码提炼出来，代码应用一目了然。

除了“查询”版块以外，“检索”版块还可以通过输入个别关键字或字母，迅速检索到包含该关键字的所有技术文档，从而提高关键字的检索速度及检索的深度。

5.8.2　使用英语词典及其他辅助词典协助编程

我是一名即将毕业的学生，准备做一个“学生成绩管理系统”的程序作为自己的毕业作品。由于没有编程经验，加上学到的内容有限，对 SQL

更多资源请访问稀酷客(www.ckook.com)

语句、术语、常用的英语解释等这些知识了解得甚少，所以麻烦也就接踵而来。

在开发过程中由于我对代码的规范不太了解，所以代码的书写就很不规范（如多行代码写在一行、多数代码没有注释等），如下段代码所示。结果在中途检查程序开发进度的时候，老师说我的程序不易于阅读、代码不一目了然，希望我重新修改。

```
CREATE TRIGGER trigger_name
ON { table | view }
{
{ { FOR | AFTER | INSTEAD OF } { [DELETE] [,] [INSERT] [,] [UPDATE] }
AS
sql_statement [ ...n ]
}
}
Create trigger selectdate
on 并台记录表
after insert
as
select * from 并台记录表
```

不规范的程序代码

焦头烂额之际，我想到了《C#编程全能词典》软件，希望通过其中的“编程规范”辅助词典帮我解决该问题。

在“辅助词典”首界面的左侧目录列出了所有语言的编码规范，根据需要我选择了“C#”/“C#统一规范”文档名称，右侧显示了关于编码规范的具体内容。效果如图 5.47 所示。

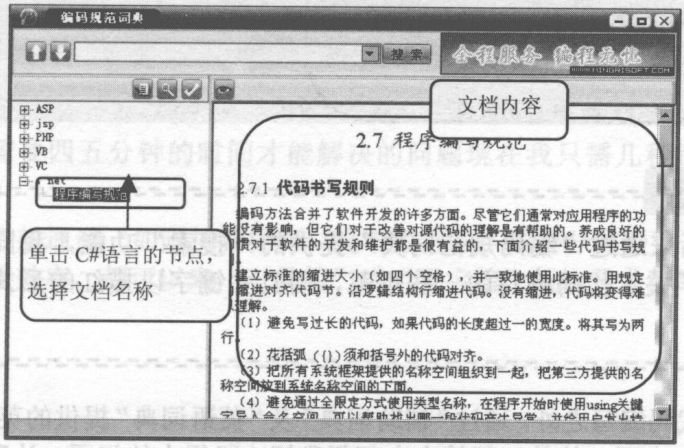


图 5.47　浏览编码规范文档浏览

为了方便阅读文档，我单击了隐藏目录按钮，隐藏了左侧目录，在文

更多资源请访问稀酷客(www.ckook.com)

档中我找到了其中介绍如何编写代码的一段文档，效果如图 5.48 所示。

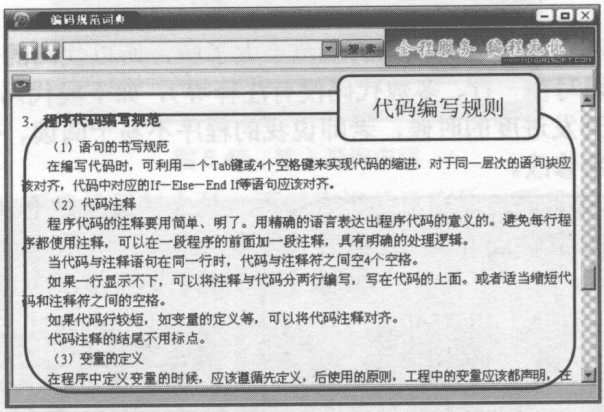


图 5.48　隐藏目录浏览代码编写规则

在仔细阅读了文档之后，我很快按照“编码规范词典”中文档的要求重新改写了代码，如下段代码所示。改写后的代码符合了老师的要求，程序也很快进入了收尾阶段。

```
CREATE TRIGGER trigger_name
ON { table | view }
{
    { { FOR | AFTER | INSTEAD OF } { [DELETE] [,] [INSERT] [,] [UPDATE] }
    AS
      sql_statement [ ...n ]
    }
}
Create trigger selectdate
on 并台记录表
after insert
as
select * from 并台记录表
```

完整规范的代码



提示

后来通过“编码规范词典”提供的“搜索”功能，我同样很快找到了关于代码编写的一些规范，而且关键字以描红的形式进行了显示。

除了代码编写之外，通过“辅助词典”及“英语词典”提供的英语、UML、SQL 语句、术语、软件工程等内容都帮我解决了很大的问题，让我的编程之路轻松无忧。

更多资源请访问稀酷客(www.ckook.com)

5.8.3 借助各种实用工具实现高效开发

程序的开发是一个漫长的过程，我是一名有多年编程经验的开发人员，开发一个程序对于我来说不是问题，但如何能节省开发时间、提高开发效率却是我现在迫切需要解决的难题。

比如程序在开发完成之后，后期通常需要进行测试，测试时会不可避免地需发附加、分离数据库等操作。每次我都需要打开 SQL Server 进行手动附加和分离，如此重复性的操作浪费了我大量的时间。现在有了《C#编程全能词典》软件，“应用”模式中“工具集”版块提供的工具帮了我大忙。

在“工具集”版块中我选择“SQL Server 数据库”工具，界面如图 5.49 所示。在文本框中我输入了想要分离的数据库名称“db_Ex01_Data.MDF”。

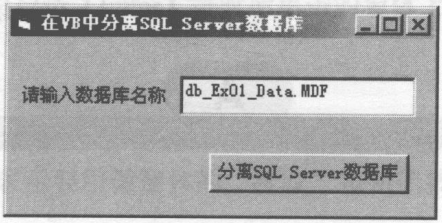


图 5.49 输入要分离的数据库名称

然后单击“分离 SQL Server 数据库”按钮，程序弹出一个提示框，提示“是否分离数据库？”，效果如图 5.50 所示。单击“是”按钮以后，数据库轻松分离。

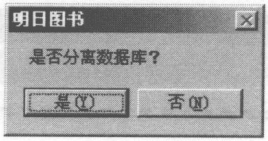


图 5.50 “是否分离数据库”提示框

通常需要四五分钟的时间才能解决的问题现在我只需几秒钟就能轻松搞定。

如果你想通过（如附加数据库、提取数据表、图表分类预览）辅助工具来提高开发效率，都可以来“工具箱”或“百宝箱”版块寻求帮助。

更多资源请访问稀酷客(www.ckook.com)

第 6 章
函数速查

A

Abort 方法

在调用此方法的线程上引发 Thread Abort Exception，以开始终止此线程的过程。



语法

```
public void Abort ()
```



说明

通常在关闭线程时调用该方法。

Abs 方法

返回指定数字的绝对值。



语法

```
public static decimal Abs(decimal value)
public static double Abs(double value)
public static short Abs(short value)
public static int Abs(int value)
public static long Abs(long value)
```

更多资源请访问稀酷客(www.ckook.com)

```
public static sbyte Abs(sbyte value)
public static float Abs(float value)
```

Abs 方法的语法中各参数的说明如表 A.1 所示。

表 A.1　Abs 方法的语法中各参数的说明

参　数	描　述
value	decimal、double、short、int、long、sbyte、float 类型的数字
返回值	返回指定数字的绝对值

AcceptButton 属性

获取或设置当用户按【Enter】键时所单击的窗体上的按钮。

语法

```
public IButtonControl AcceptButton { get; set; }
```

IButtonControl 表示要用做窗体的“接收”按钮的按钮。

示例

本示例将窗体 Form1 的“接收”按钮设置为 button1。

```
Form1 frm=new Form1();
Frm.AcceptButton=button1;
```

本示例将当前窗体的“接收”按钮设置为 button1。

```
this.AcceptButton = button1;
```

Activate 事件

当使用代码激活或用户激活窗体时发生。

语法

```
public event EventHandler Activate
```

Add 方法

语法

- ListBox 控件的 Add 方法

更多资源请访问稀酷客(www.ckook.com)

```
public int Add (Object item)
```

ListBox.Add 方法的语法中各参数的说明如表 A.2 所示。

表 A.2 ListBox.Add 方法的语法中各参数的说明

参　数	描　述
item	一个对象，表示要添加到集合中的项
返回值	集合中项的从零开始的索引

- BindingSource 控件的 Add 方法

```
public virtual int Add (Object value)
```

BindingSource.Add 方法的语法中各参数的说明如表 A.3 所示。

表 A.3 BindingSource.Add 方法的语法中各参数的说明

参　数	描　述
value	添加到内部列表中的 Object
返回值	从零开始的索引，value 按照该索引添加到由 List 属性表示的基础列表中

- ArrayList 类的 Add 方法

```
public virtual int Add (Object value)
```

ArrayList.Add 方法的语法中各参数的说明如表 A.4 所示。

表 A.4 ArrayList.Add 方法的语法中各参数的说明

参　数	描　述
value	要添加到 ArrayList 末尾处的 Object，该值可以为空引用
返回值	ArrayList 索引，已在此处添加了 value

- ListView 控件的 Add 方法

```
public virtual ListViewItem Add (string text,int imageIndex)
```

ListView.Add 方法的语法中各参数的说明如表 A.5 所示。

表 A.5 ListView.Add 方法的语法中各参数的说明

参　数	描　述
text	项的文本
imageIndex	要为该项显示的图像的索引
返回值	已添加到集合中的 ListViewItem

- Workbooks 对象的 Add 方法

```
Workbook Add(
    [In, Optional] object Template);
```

Workbooks.Add 方法的语法中各参数的说明如表 A.6 所示。

更多资源请访问稀酷客(www.ckook.com)

表 A.6　Workbooks.Add 方法的语法中各参数的说明

参　数	描　述
Template	指定是否添加工作簿，如果为 true，表示添加；否则，表示不添加
返回值	返回 Workbook 类型，表示一个 Excel 工作簿

AddDays 方法

将指定的天数加到此实例的值上。

语法

```
public DateTime AddDays(double value)
```

AddDays 方法的语法中各参数的说明如表 A.7 所示。

表 A.7　AddDays 方法的语法中各参数的说明

参　数	描　述
value	由整数和小数部分组成的天数。value 参数可以是负数也可以是正数
返回值	DateTime，其值是此实例所表示的日期和时间与 value 所表示的天数之和

示例

本示例主要实现“取得当前日期的前一天”功能，使用了 DateTime 结构的 AddDays 方法，在指定的 DateTime 实例的值上加上“–1”天取得所要的结果，代码如下：

```
TextBox1.Text = DateTime.Now.AddDays(-1).ToString();
```

AddString 方法

向 GraphicsPath 路径中添加文本字符串。

语法

```
public void AddString (
    string s,
    FontFamily family,
    int style,
    float emSize,
    PointF origin,
    StringFormat format)
```

AddString 方法的语法中各参数的说明如表 A.8 所示。

更多资源请访问稀酷客(www.ckook.com)

表 A.8 AddString 方法的语法中各参数的说明

参　数	描　述
s	要添加的文本字符串
family	一个 FontFamily，表示绘制文本所用字体的名称
style	一个 FontStyle 枚举，它表示有关文本的样式信息（粗体、斜体等）。它必须为整数
emSize	限定字符的 Em（字体大小）方框的高度
origin	一个 PointF，它表示文本从其起始的点
format	指定文本格式设置信息（如行间距和对齐方式）的 StringFormat

AllowEdit 属性

获取一个值，该值指示是否可以编辑基础列表中的项。



语法

```
public virtual bool AllowEdit { get; }
```

属性值：为 true 表示列表项可以编辑；为 false 表示列表不可以编辑。

AppendText 方法

向文本框的当前文本追加文本。



语法

```
public void AppendText (string text)
```

AppendText 方法的语法中各参数的说明如表 A.9 所示。

表 A.9 AppendText 方法的语法中各参数的说明

参　数	描　述
text	要向文本框的当前内容追加的文本

Appllication 类

Appllication 类提供静态方法和属性以管理应用程序，例如启动和停止应用程序、处理 Windows 消息的方法和获取应用程序信息的属性，无法继承此类。



属性

Application 类的常用属性及描述如表 A.10 所示。

更多资源请访问稀酷客(www.ckook.com)

表 A.10　Application 类的常用属性及描述

属　性	描　述
AllowQuit	获取指示调用方能否退出该应用程序的值
CommonAppDataPath	获取所有用户共享的应用程序数据的路径
CommonAppDataRegistry	获取所有用户共享的应用程序数据的注册表项
CompanyName	获取与该应用程序关联的公司名称
CurrentCulture	获取或设置当前线程的区域性信息
CurrentInputLanguage	获取或设置当前线程的当前输入语言
ExecutablePath	获取启动了应用程序的可执行文件的路径，包括可执行文件的名称
OpenForms	获取属于应用程序的打开窗体的集合
ProductName	获取与该应用程序关联的产品名称
ProductVersion	获取与该应用程序关联的产品版本
StartupPath	获取启动了应用程序的可执行文件的路径，不包括可执行文件的名称
UserAppDataPath	获取用户的应用程序数据的路径
UserAppDataRegistry	获取用户的应用程序数据的注册表项
UseWaitCursor	获取或设置是否将等待光标用于应用程序的所有打开的窗体
VisualStyleState	获取指定可视样式如何应用于应用程序窗口的值



方法

Application 类的常用方法及描述如表 A.11 所示。

表 A.11　Application 类的常用方法及描述

方　法	描　述
AddMessageFilter	添加消息筛选器以便在向目标传送 Windows 消息时监视这些消息
EnableVisualStyles	启用应用程序的可视样式
Exit	通知所有消息泵必须终止，并且在处理消息以后关闭所有应用程序窗口
ExitThread	退出当前线程上的消息循环，并关闭该线程上的所有窗口
FilterMessage	对窗口消息运行任何筛选器，并返回已修改消息的副本
RegisterMessageLoop	注册一个回调以检查消息循环是否正在宿主环境中运行
RemoveMessageFilter	从应用程序的消息泵移除一个消息筛选器
Restart	关闭应用程序并立即启动新的实例
Run	在当前线程上开始运行标准应用程序消息循环
SetSuspendState	挂起系统或使系统休眠，或者请求系统挂起或休眠
UnregisterMessageLoop	注销使用 RegisterMessageLoop 发出的消息循环回调

ArrayList 类

ArrayList 类主要用于对一个数组中的元素进行各种处理，它的使用大小可按需动态增加的数组实现 IList 接口。

更多资源请访问稀酷客(www.ckook.com)

属性

ArrayList 类的常用属性及描述如表 A.12 所示。

表 A.12 ArrayList 类的常用属性及描述

属　性	描　述
Capacity	获取或设置 ArrayList 可包含的元素数
Count	获取 ArrayList 中实际包含的元素数
IsFixedSize	获取一个值，该值表示 ArrayList 是否具有固定大小
IsReadOnly	获取一个值，该值表示 ArrayList 是否为只读
Item	获取或设置指定索引处的元素

方法

ArrayList 类的常用方法及描述如表 A.13 所示。

表 A.13 ArrayList 类的常用方法及描述

方　法	描　述
Add	将对象添加到 ArrayList 的结尾处
AddRange	将 ICollection 的元素添加到 ArrayList 的末尾
Clear	从 ArrayList 中移除所有元素
Contains	确定某元素是否在 ArrayList 中
CopyTo	将 ArrayList 或它的一部分复制到一维数组中
FixedSize	返回具有固定大小的列表包装，其中的元素允许修改，但不允许添加或移除
GetEnumerator	返回循环访问 ArrayList 的枚举数
GetRange	返回 ArrayList，它表示源 ArrayList 中元素的子集
IndexOf	返回 ArrayList 或它的一部分中某个值的第一个匹配项的从零开始的索引
Insert	将元素插入 ArrayList 的指定索引处
InsertRange	将集合中的某个元素插入 ArrayList 的指定索引处
LastIndexOf	返回 ArrayList 或它的一部分中某个值的最后一个匹配项的从零开始的索引
Remove	从 ArrayList 中移除特定对象的第一个匹配项
RemoveAt	移除 ArrayList 的指定索引处的元素
RemoveRange	从 ArrayList 中移除一定范围的元素
Repeat	返回 ArrayList，它的元素是指定值的副本
Reverse	将 ArrayList 或它的一部分中元素的顺序反转
SetRange	将集合中的元素复制到 ArrayList 中一定范围的元素上
Sort	对 ArrayList 或它的一部分中的元素进行排序
ToArray	将 ArrayList 的元素复制到新数组中

更多资源请访问稀酷客(www.ckook.com)

as 操作符

as 操作符用于转换引用类型。



说明

as 操作符类似于强制转换，所不同的是，当转换失败时，操作符将产生空值，而不是引发异常。



示例

本示例使用 as 操作符将一个 object 类型的变量转换为 string 类型。

```
Object someObject;
string s = someObject as string;
```

本示例使用 as 操作符将一个 int 类型的变量转换为 string 类型。

```
int i=1;s
string s = i as string;
```

ASCII 码

ASCII 码作为一种键盘表示方式，在程序中经常用到，编程中经常用到的 ASCII 码如表 A.14 所示。

表 A.14　编程中经常使用的 ASCII 码

键　盘	ASCII 码	键　盘	ASCII 码
0	48	1	49
2	50	3	51
4	52	5	53
6	54	7	55
8	56	9	57
a	97	b	98
c	99	d	100
e	101	f	102
g	103	h	104
i	105	j	106
k	107	l	108
m	109	n	110

更多资源请访问稀酷客(www.ckook.com)

续表

键　盘	ASCII 码	键　盘	ASCII 码
O	111	p	112
q	113	r	114
s	115	t	116
u	117	v	118
w	119	x	120
y	121	z	122
A	65	B	66
C	67	D	68
E	69	F	70
G	71	H	72
I	73	J	74
K	75	L	76
M	77	N	78
O	79	P	80
Q	81	R	82
S	83	T	84
U	85	V	86
W	87	X	88
Y	89	Z	90

ASCIIEncoding 类

表示 Unicode 字符的 ASCII 字符编码。

属性

ASCIIEncoding 类的常用属性及描述如表 A.15 所示。

表 A.15　ASCIIEncoding 类的常用属性及描述

属　性	描　述
BodyName	在派生类中重写时，获取可与邮件代理正文标记一起使用的当前编码的名称
EncodingName	在派生类中重写时，获取当前编码的可读说明
HeaderName	在派生类中重写时，获取可与邮件代理标题标记一起使用的当前编码的名称
IsReadOnly	在派生类中重写时，获取一个值，该值指示当前的编码是否为只读

更多资源请访问稀酷客(www.ckook.com)

方法

ASCIIEncoding 类的常用方法及描述如表 A.16 所示。

表 A.16　ASCIIEncoding 类的常用方法及描述

方　法	描　述
GetByteCount	计算对一组字符进行编码时产生的字节数
GetBytes	将一组字符编码为一个字节序列
GetCharCount	计算对一个字节序列进行解码所产生的字符数
GetChars	将一个字节序列解码为一组字符
GetString	将一个字节序列解码为一个字符串

Attributes 属性

获取和设置文件的属性。

语法

```
public FileAttributes Attributes { get; set; }
```

Attributes 属性的成员及描述如表 A.17 所示。

表 A.17　Attributes 属性的成员及描述

成　员	描　述
ReadOnly	文件为只读
Hidden	文件是隐藏的，因此没有包括在普通的目录列表中
System	文件为系统文件。文件是操作系统的一部分或由操作系统以独占方式使用
Directory	文件为一个目录
Archive	文件的存档状态。应用程序使用此属性为文件加上备份或移除标记
Device	保留供将来使用
Normal	文件正常，没有设置其他的属性。此属性仅在单独使用时有效
Temporary	文件是临时文件。文件系统试图将所有数据保留在内存中以便更快地访问，而不是将数据刷新回大容量存储器中。不再需要临时文件时，应用程序会立即将其删除
SparseFile	文件为稀疏文件。稀疏文件一般是数据通常为零的大文件
ReparsePoint	文件包含一个重新分析点，它是一个与文件或目录关联的用户定义的数据块
Compressed	文件已压缩
Offline	文件已脱机。文件数据不能立即供使用
NotContentIndexed	操作系统的内容索引服务不会创建此文件的索引
Encrypted	该文件或目录是加密的。对于文件来说，表示文件中的所有数据都是加密的。对于目录来说，表示新创建的文件和目录在默认情况下是加密的

更多资源请访问稀酷客(www.ckook.com)

AVG 聚合函数

返回组中值的平均值，空值将被忽略。

语法

```
AVG ([ALL|DISTINCT] expression )
```

AVG 聚合函数的语法中各参数的说明如表 A.18 所示。

表 A.18　AVG 聚合函数的语法中各参数的说明

参　数	描　述
ALL	对所有的值进行聚合函数运算。ALL 是默认设置
DISTINCT	指定 AVG 操作只使用每个值的唯一实例，而不管该值出现了多少次
expression	精确数字或近似数字数据类型类别的表达式（bit 数据类型除外），不允许使用聚合函数和子查询

示例

本示例在某班的基础信息表（tb_stu）中，利用聚合函数 AVG 查询统计本班学生的平均年龄。

```
select avg(年龄) as 平均年龄 from tb_stu
```

本示例在学生成绩表中，利用聚合函数 AVG 计算“小王”的平均成绩。

```
select avg(成绩) as 平均成绩 from 学生成绩表 where name='小王' order by 成绩
```


更多资源请访问稀酷客(www.ckook.com)

B

BackColor 属性

获取或设置控件的背景色。

语法

```
public virtual Color BackColor { get; set; }
```

属性值：表示控件背景色的 Color，默认为 DefaultBackColor 属性的值。

BackgroudColor 属性

获取或设置 DataGridView 控件的背景色。

语法

```
public Color BackgroundColor { get; set; }
```

属性值：一个 Color，表示 DataGridView 的背景色，默认为 SystemColors.AppWorkspace。

示例

本示例将 dataGridView1 控件的背景颜色设置为蓝色。

```
dataGridView1.BackgroundColor = Color.Blue;
```

本示例将 dataGridView1 控件的背景颜色设置为白色。

```
dataGridView1.BackgroundColor = Color.White;
```

BackgroudImage 属性

获取或设置在控件中显示的背景图像。

语法

```
public virtual Image BackgroundImage { get; set; }
```

属性值：一个 Image，它表示在控件的背景中显示的图像。

更多资源请访问稀酷客(www.ckook.com)

示例

本示例将 button1 控件的背景图片设置为“明日科技.bmp”。

```
    this.button1.BackgroundImage = System.Drawing.Image.FromFile("明日科技.bmp");
```

本示例将 panel1 控件的背景图片设置为“明日科技.bmp”。

```
    this. panel1.BackgroundImage = System.Drawing.Image.FromFile("明日科技.bmp");
```

本示例将 listbox1 控件的背景图片设置为“明日科技.bmp”。

```
    this. listbox1.BackgroundImage = System.Drawing.Image.FromFile("明日科技.bmp");
```

本示例将当前窗体的背景图片设置为“明日科技.bmp”。

```
    private void Form1_Load(object sender, EventArgs e)
    {
        this.BackgroundImage = System.Drawing.Image.FromFile("明日科技.bmp");
    }
```

BackgroundWorker 组件

运行程序时，有些操作可能需要很长的执行时间，例如：图像下载、Web 服务调用和文件下载、上传等，这些操作可能导致用户界面在操作运行时挂起，如果此时需要用户界面的响应，可以使用 BackgroundWorker 组件，在不同于应用程序的主用户界面线程的另一线程上异步（在后台）执行耗时的操作。

属性

BackgroundWorker 组件的常用属性及描述如表 B.1 所示。

表 B.1　BackgroundWorker 组件的常用属性及描述

属　性	描　述
CancellationPending	获取一个值，指示应用程序是否已请求取消后台操作
Events	获取附加到此 BackgroundWorker 的事件处理程序的列表
IsBusy	获取一个值，指示 BackgroundWorker 是否正在运行异步操作
WorkerReportsProgress	获取或设置一个值，该值指示 BackgroundWorker 能否报告进度更新
WorkerSupportsCancellation	获取或设置一个值，该值指示 BackgroundWorker 是否支持异步取消

更多资源请访问稀酷客(www.ckook.com)

方法

BackgroundWorker 组件的常用方法及描述如表 B.2 所示。

表 B.2　BackgroundWorker 组件的常用方法及描述

方　法	描　述
CancelAsync	请求取消挂起的后台操作
OnDoWork	引发 DoWork 事件
OnRunWorkerCompleted	引发 RunWorkerCompleted 事件
RunWorkerAsync	开始执行后台操作

事件

BackgroundWorker 组件的常用事件及描述如表 B.3 所示。

表 B.3　BackgroundWorker 组件的常用事件及描述

事　件	描　述
DoWork	调用 RunWorkerAsync 时发生
RunWorkerCompleted	当后台操作已完成、被取消或引发异常时发生

base 关键字

功能

用于从派生类中访问基类的成员，其使用范围如下：

（1）调用基类上已被其他方法重写的方法。

（2）指定创建派生类实例时应调用的基类构造函数。

注意

在静态方法中使用 base 关键字是错误的。

示例

本示例讲解如何在派生类中使用 base 关键字调用基类中的公有字段。

```
Class BaseClass
{
    public int a=1;
}
class MyClass:BaseClass
```


更多资源请访问稀酷客(www.ckook.com)

```
{
     public int result()
{
    return base.a;
}
}
```

BeginConnect 方法

开始一个对远程主机连接的异步请求，主机由 IP Address 和端口号指定。

语法

```
public IAsyncResult BeginConnect
(
    IPAddress address,
    int port,
    AsyncCallback requestCallback,
    Object state)
```

BeginConnect 方法的语法中各参数的说明如表 B.4 所示。

表 B.4 BeginConnect 方法的语法中各参数的说明

参　数	描　述
address	远程主机的 IPAddress
port	远程主机的端口号
requestCallback	一个 AsyncCallback 委托，它引用连接操作完成时要调用的方法
state	一个用户定义对象，其中包含连接操作的相关信息。当操作完成时，此对象会被传递给 requestCallback 委托
返回值	IAsyncResult，它引用异步连接

命名空间

System.Net.Sockets

BeginEdit 方法

用于将当前单元格置于编辑模式下。

语法

```
public virtual bool BeginEdit (bool selectAll)
```

BeginEdit 方法的语法中各参数的说明如表 B.5 所示。

更多资源请访问稀酷客(www.ckook.com)

表 B.5 BeginEdit 方法的语法中各参数的说明

参　数	描　述
selectAll	如果选择该单元格的所有内容，为 true；如果不选择任何内容，则为 false
返回值	如果当前的单元格已经处于编辑模式或者成功进入编辑模式，则为 true；否则为 false

BinaryReader 类

用特定的编码将基元数据类型读做二进制值。

构造函数

```
public BinaryReader(Stream input);
public BinaryReader(Stream input,Encoding encoding);
```

BinaryReader 类的构造函数中各参数的说明如表 B.6 所示。

表 B.6 BinaryReader 类的构造函数中各参数的说明

参　数	描　述
input	提供的流
encoding	字符编码

命名空间

System.IO

属性

BinaryReader 类的常用属性及描述如表 B.7 所示。

表 B.7 BinaryReader 类的常用属性及描述

属　性	描　述
BaseStream	公开对 BinaryReader 的基础流的访问

方法

BinaryReader 类的常用方法及描述如表 B.8 所示。

表 B.8 BinaryReader 类的常用方法及描述

方　法	描　述
Close	关闭当前阅读器及基础流
Dispose	释放由 BinaryReader 占用的非托管资源，还可以另外再释放托管资源
FillBuffer	用从流中读取的指定字节数填充内部缓冲区

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
PeekChar	返回下一个可用的字符，并且不提升字节或字符的位置
Read	从基础流中读取字符，并提升流的当前位置
ReadBoolean	从当前流中读取 Boolean 值，并使该流的当前位置提升一个字节
ReadByte	从当前流中读取下一个字节，并使流的当前位置提升一个字节
ReadBytes	从当前流中将 count 个字节读入字节数组，并使当前位置提升 count 个字节
ReadChar	从当前流中读取下一个字符，并根据所使用的 Encoding 和从流中读取的特定字符，提升流的当前位置
ReadChars	从当前流中读取 count 个字符，以字符数组的形式返回数据，并根据所使用的 Encoding 和从流中读取的特定字符，提升当前位置
ReadDecimal	从当前流中读取十进制数值，并将该流的当前位置提升 16 个字节
ReadDouble	从当前流中读取 8 字节浮点值，并使流的当前位置提升 8 个字节
ReadInt16	从当前流中读取 2 字节有符号整数，并使流的当前位置提升 2 个字节
ReadInt32	从当前流中读取 4 字节有符号整数，并使流的当前位置提升 4 个字节
ReadInt64	从当前流中读取 8 字节有符号整数，并使流的当前位置向前移动 8 个字节
ReadSByte	从此流中读取 1 个有符号字节，并使流的当前位置提升 1 个字节
ReadSingle	从当前流中读取 4 字节浮点值，并使流的当前位置提升 4 个字节
ReadString	从当前流中读取一个字符串。字符串有长度前缀，一次 7 位地被编码为整数
ReadUInt16	使用 Little-Endian 编码从当前流中读取 2 字节无符号整数，并将流的位置提升 2 个字节
ReadUInt32	从当前流中读取 4 字节无符号整数，并使流的当前位置提升 4 个字节
ReadUInt64	从当前流中读取 8 字节无符号整数，并使流的当前位置提升 8 个字节

BinaryWriter 类

以二进制形式将基元类型写入流，并支持用特定的编码写入字符串。



构造函数

```
protected BinaryWriter()
public BinaryWriter(Stream output);
public BinaryWriter(Stream output,Encoding encoding);
```

BinaryWriter 类的构造函数中各参数的说明如表 B.9 所示。

表 B.9　BinaryWriter 类的构造函数中各参数的说明

参　数	描　述
input	提供的流
encoding	字符编码

更多资源请访问稀酷客(www.ckook.com)

命名空间

System.IO

属性

BinaryWriter 类的常用属性及描述如表 B.10 所示。

表 B.10　BinaryWriter 类的常用属性及描述

属　性	描　述
BaseStream	获取 BinaryWriter 的基础流

方法

BinaryWriter 类的常用方法及描述如表 B.11 所示。

表 B.11　BinaryWriter 类的常用方法及描述

方　法	描　述
Close	关闭当前的 BinaryWriter 和基础流
Dispose	释放由 BinaryWriter 占用的非托管资源，还可以另外再释放托管资源
Flush	清理当前编写器的所有缓冲区，使所有缓冲数据写入基础设备
Seek	设置当前流中的位置
Write	将值写入当前流

BindingNavigator 控件

BindingNavigator 控件是绑定到数据的控件的导航和操作用户界面(UI)。使用 BindingNavigator 控件，用户可以在 Windows 窗体中导航和操作数据。

属性

BindingNavigator 控件的常用属性及描述如表 B.12 所示。

表 B.12　BindingNavigator 控件的常用属性及描述

属　性	描　述
AddNewItem	获取或设置表示“新添”按钮的 ToolStripItem
BindingContext	获取或设置 BindingNavigator 的绑定上下文
BindingSource	获取或设置 System.Windows.Forms.BindingSource 组件，它是数据的来源
CountItem	获取或设置 ToolStripItem，它显示关联的 BindingSource 中的总项数
CountItemFormat	获取或设置用于设置在 CountItem 控件中显示的信息的格式的字符串
Enabled	获取或设置一个值，该值指示控件是否可以对用户交互作出响应

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
Items	获取属于 BindingNavigator 的所有项
MoveFirstItem	获取或设置与“移到第一条记录”功能关联的 ToolStripItem
MoveLastItem	获取或设置与“移到最后”功能关联的 ToolStripItem
MoveNextItem	获取或设置与“移到下一条记录”功能关联的 ToolStripItem
MovePreviousItem	获取或设置与“移到上一条记录”功能关联的 ToolStripItem
Name	获取或设置控件的名称
Orientation	获取 ToolStripPanel 的方向
PositionItem	获取或设置 ToolStripItem，它显示 BindingSource 中的当前位置
Text	获取或设置与此控件关联的文本
TextDirection	获取或设置在 BindingNavigator 上绘制文本的方向
Visible	获取或设置一个值，该值指示是否显示该控件及其所有父控件

方法

BindingNavigator 控件的常用方法及描述如表 B.13 所示。

表 B.13　BindingNavigator 控件的常用方法及描述

方　法	描　述
BringToFront	将控件带到 Z 顺序的前面
GetNextItem	通过从指定参考点沿指定方向移动来检索下一个 ToolStripItem
ProcessCmdKey	处理命令键
SendToBack	将控件发送到 Z 顺序的后面

事件

BindingNavigator 控件的常用事件及描述如表 B.14 所示。

表 B.14　BindingNavigator 控件的常用事件及描述

事　件	描　述
ItemClicked	在单击 ToolStripItem 时发生
ItemRemoved	当从 ToolStripItemCollection 中移除 ToolStripItem 时发生

Bitmap 类

Bitmap 是用于处理由像素数据定义的图像的对象。

命名空间

System.Drawing

更多资源请访问稀酷客(www.ckook.com)

属性

Bitmap 类的常用属性及描述如表 B.15 所示。

表 B.15 Bitmap 类的常用属性及描述

属　性	描　述
Height	获取此 Image 的高度（以像素为单位）
HorizontalResolution	获取此 Image 的水平分辨率（以“像素/英寸”为单位）
PixelFormat	获取此 Image 的像素格式
RawFormat	获取此 Image 的文件格式
Size	获取此图像的以像素为单位的宽度和高度
VerticalResolution	获取此 Image 的垂直分辨率（以“像素/英寸”为单位）
Width	获取此 Image 的宽度（以像素为单位）

方法

Bitmap 类的常用方法及描述如表 B.16 所示。

表 B.16 Bitmap 类的常用方法及描述

方　法	描　述
Clone	创建此 Bitmap（用指定的 PixelFormat 定义）部分的副本
GetPixel	获取此 Bitmap 中指定像素的颜色
GetThumbnailImage	返回此 Image 的缩略图
MakeTransparent	使默认的透明颜色对此 Bitmap 透明
RotateFlip	此方法旋转、翻转或者同时旋转和翻转 Image
Save	将该 Image 保存到指定的文件或流
SetPixel	设置此 Bitmap 中指定像素的颜色
SetResolution	设置此 Bitmap 的分辨率

Border 属性

获取一个表示 Button 的边框的 Border。

语法

```
public virtual Border Border { get; }
```

属性值：一个 Border，表示 Button 的边框。

命名空间

Microsoft.Office.Tools.Excel.Controls

更多资源请访问稀酷客(www.ckook.com)

BorderStyle 属性

此属性指示控件的边框样式。

语法

```
public BorderStyle BorderStyle { get; set; }
```

BorderStyle 属性的枚举值及描述如表 B.17 所示。

表 B.17 BorderStyle 属性的枚举值及描述

枚举值	描述
Fixed3D	三维边框
FixedSingle	单行边框
None	无边框

示例

本示例设置 label1 控件的边框样式为单行边框。

```
this. label1.BorderStyle = BorderStyle. FixedSingle;
```

本示例设置 panel1 控件的边框样式为无边框。

```
this. panel1.BorderStyle = BorderStyle. None;
```

本示例设置 dataGridView1 控件的边框样式为三维边框。

```
this.dataGridView1.BorderStyle = BorderStyle.Fixed3D;
```

break 语句

功能

break 语句可以用来终止一条迭代语句，导致控制流程跳转到迭代语句的下一条语句。break 可以放在循环体的任意地方来实现这一目的。

说明

若要中止整个嵌套循环，可以使用 return 语句。

BringToFront 方法

将控件带到 Z 顺序的前面。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public void BringToFront()
```

说明

如果控件是另一个控件的子控件，则子控件移到 Z 顺序的前面。BringToFront 不会使一个控件成为顶级控件，并且不会引发 Paint 事件。

Button 控件

Button 控件允许用户通过单击来执行各种操作。当 Button 控件被单击时，它看起来像是被按下，然后被释放。每当用户单击 Button 控件时，即调用其 Click 事件执行各种操作。

属性

Button 控件的常用属性及描述如表 B.18 所示。

表 B.18 Button 控件的常用属性及描述

属　性	描　述
BackColor	获取或设置控件的背景色
BackgroundImage	获取或设置在控件中显示的背景图像
Controls	获取包含在控件内的控件的集合
Cursor	获取或设置当鼠标指针位于控件上时显示的光标
Dock	获取或设置哪些控件边框停靠到其父控件并确定控件如何随其父级一起调整大小
Enabled	获取或设置一个值，该值指示控件是否可以对用户交互作出响应
FlatStyle	获取或设置按钮控件的平面样式外观
Font	获取或设置控件显示的文字的字体
ForeColor	获取或设置控件的字体颜色
ImageAlign	获取或设置按钮控件上的图像对齐方式
Name	获取或设置控件的名称
Tag	获取或设置包含有关控件的数据的对象
Text	获取或设置与此控件关联的文本
Visible	获取或设置一个值，该值指示是否显示该控件及其所有父控件
Width	获取或设置控件的宽度
Height	获取或设置控件的高度

更多资源请访问稀酷客(www.ckook.com)

方法

Button 控件的常用方法及描述如表 B.19 所示。

表 B.19 Button 控件的常用方法及描述

方 法	描 述
BringToFront	将控件带到 Z 顺序的前面
Focus	为控件设置输入焦点
Hide	对用户隐藏控件
Show	向用户显示控件

事件

Button 控件的常用事件及描述如表 B.20 所示。

表 B.20 Button 控件的常用事件及描述

事 件	描 述
Click	在单击控件时发生
KeyPress	在控件有焦点的情况下释放键时发生
KeyDown	在控件有焦点的情况下按下键时发生
KeyUp	在控件有焦点的情况下按上键时发生
MouseDown	当鼠标指针位于控件上并按下鼠标键时发生
MouseUp	在鼠标指针位于控件上并按上鼠标键时发生

变量

功能

变量本身被用来存储特定类型的数据，可以根据需要随时改变变量中所存储的数据值。变量具有名称、类型和值。变量名是变量在程序源代码中的标识；变量类型确定它所代表的内存的大小和类型；变量值是指它所代表的内存块中的数据。在程序的执行过程中，变量的值可以发生变化。使用变量之前必须先声明变量，即指定变量的类型和名称。

类型

C#中的变量类型根据其定义可以分为两种：一种是值类型，另一种是引用类型，这两种类型的差异在于数据的存储方式，值类型的变量本身直接存储数据；而引用类型则存储实际数据的引用，程序通过此引用找到真正的数据。

更多资源请访问稀酷客(www.ckook.com)

- 值类型

值类型变量直接存储其数据值，主要包含整数类型、浮点类型，以及布尔类型等。值类型变量在堆栈中进行分配，因此效率很高，使用值类型的主要目的是为了提高性能。值类型具有如下特性。

（1）值类型变量都存储在堆栈中；

（2）访问值类型变量时，一般都是直接访问其实例；

（3）每个值类型变量都有自己的数据副本，因此对一个值类型变量的操作不会影响其他变量；

（4）复制值类型变量时，复制的是变量的值，而不是变量的地址；

（5）值类型变量不能为 null，必须具有一个确定的值。

- 引用类型

引用类型是构建 C#应用程序的主要对象类型数据，在应用程序执行的过程中，预先定义的对象类型以 new 创建对象实例，并且存储在堆栈中。引用类型具有如下特征。

（1）必须在托管堆中为引用类型变量分配内存；

（2）必须使用 new 关键字来创建引用类型变量；

（3）在托管堆中分配的每个对象都有与之相关联的附加成员，这些成员必须被初始化；

（4）引用类型变量是由垃圾回收机制来管理的；

（5）多个引用类型变量可以引用同一对象，这种情形下，对一个变量的操作会影响另一个变量所引用的同一对象；

（6）引用类型被赋值前的值都是 null。

声明

一个变量是由一个类型和跟在后面的一个或多个变量名组成的，多个变量之间用逗号分开，声明变量以分号结束。

声明变量时，要注意变量名的命名规则。C#的变量名是一种标识符，应该符合标识符的命名规则。变量名是区分大小写的，下面列出变量的命名规则。

- 变量名只能由数字、字母和下画线组成。
- 变量名的第一个符号只能是字母和下画线，不能是数字。
- 不能使用关键字作为变量名。
- 一旦在一个语句块中定义了一个变量名，那么在变量的作用域内都不能再定义同名的变量。

更多资源请访问稀酷客(www.ckook.com)

作用域

变量的作用域就是可以访问该变量的代码区域，一般情况下，可以通过以下规则确定变量的作用域。

- 只要字段所属的类在某个作用域内，其字段也在该作用域内。
- 局部变量存在于表示声明该变量的块语句或方法结束的封闭花括号之前的作用域内。
- 在 for、while 或类似语句中声明的局部变量存在于该循环体内。

更多资源请访问稀酷客(www.ckook.com)

CalendarFont 属性

获取或设置应用于日历的字体样式。



语法

```
public Font CalendarFont { get; set; }
```

属性值：表示应用于日历的字体样式的 Font。



示例

本示例设置 dateTimePicker1 日历控件的字体样式。

```
    dateTimePicker1.CalendarFont = new Font("Courier New", 8.25F,
FontStyle.Italic, GraphicsUnit.Point, ((Byte)(0)));
```

CalendarForeColor 属性

获取或设置日历的前景色。



语法

```
public Color CalendarForeColor { get; set; }
```

属性值：表示日历前景色的 Color。



示例

本示例设置 dateTimePicker1 日历控件的前景色为白色。

```
dateTimePicker1.CalendarForeColor = Color.White;
```

CancelButton 属性

获取或设置当用户按 Esc 键时单击的按钮。



语法

```
public IButtonControl CancelButton { get; set; }
```


更多资源请访问稀酷客(www.ckook.com)

属性值：IButtonControl，表示窗体的“取消”按钮。

CancelEdit 属性

获取或设置一个值，该值指示是否应取消对 ListViewItem 的标签所做的更改。

语法

```
public bool CancelEdit { get; set; }
```

属性值：如果应取消 ListViewItem 标签的编辑操作，则为 true；否则为 false。

示例

本示例在编辑完 ListView 控件中的项时取消当前编辑文本。

```
    private void listView1_AfterLabelEdit(object sender,
LabelEditEventArgs e)
    {
        e.CancelEdit = true;
    }
```

CanFocus 属性

获取一个值，该值指示控件是否可以接收焦点。

语法

```
public bool CanFocus { get; }
```

属性值：如果控件可以接收焦点，则为 true；否则为 false。

CanPaste 方法

确定是否可以粘贴指定数据格式的剪贴板信息。

语法

```
public bool CanPaste (Format clipFormat)
```

CanPaste 方法的语法中各参数的说明如表 C.1 所示。

更多资源请访问稀酷客(www.ckook.com)

表 C.1　CanPaste 方法的语法中各参数的说明

参　数	描　述
clipFormat	System.Windows.Forms.DataFormats. Format 值之一
返回值	如果可以粘贴指定数据格式的剪贴板数据，则为 true；否则为 false

Capture 属性

获取或设置一个值，该值指示控件是否已捕获鼠标。



语法

```
public bool Capture { get; set; }
```

属性值：如果控件已捕获鼠标，则为 true；否则为 false。

case 语句

用来将某个表达式与一组简单表达式进行比较以确定结果。



语法

```
CASE input_expression
   WHEN when_expression THEN result_expression
      [ ...n ]
   [
      ELSE else_result_expression
END
```

case 语句的语法中各参数的说明如表 C.2 所示。

表 C.2　case 语句的语法中各参数的说明

参　数	描　述
input_expression	使用简单 CASE 格式时所计算的表达式。Input_expression 是任何有效的 SQL Server 表达式
when_expression	使用简单 CASE 格式时 input_expression 所比较的简单表达式。when_expression 是任意有效的 SQL Server 表达式。Input_expression 和每个 when_expression 的数据类型必须相同，或者是隐性转换
n	占位符，表明可以使用多个 WHEN when_expression THEN result_expression 子句或 WHEN Boolean_expression THEN result_expression 子句
result_expression	当 input_expression=when_expression 取值为 True，或者 Boolean_expression 取值为 True 时返回的表达式。result-expression 是任意有效的 SQL Server 表达式

更多资源请访问稀酷客(www.ckook.com)

续表

参　数	描　述
else_result_expression	当比较运算取值不为 True 时返回的表达式。如果省略此参数并且比较运算取值不为 True，CASE 将返回 NULL 值。else_result_expression 是任意有效的 SQL Server 表达式。else_result_expression 和所有 result_expression 的数据类型必须相同，或者必须是隐性转换

CAST 函数

将某种数据类型的表达式显式转换为另一种数据类型。



语法

```
CAST ( expression AS data_type )
```

CAST 函数的语法中各参数的说明如表 C.3 所示。

表 C.3　CAST 函数的语法中各参数的说明

参　数	描　述
expression	任何有效的 SQL Server 表达式
data_type	目标系统所提供的数据类型，包括 bigint 和 sql_variant。不能使用用户定义的数据类型

Cell 对象

表示 Word 文档中的单元格。



命名空间

Microsoft.Office.Tools.Word

CellClick 事件

在单元格的任何部分被单击时发生。



语法

```
public event DataGridViewCellEventHandler CellClick
```


更多资源请访问稀酷客(www.ckook.com)

CellEnter 事件

在DataGridView控件中的当前单元格更改或者该控件接收到输入焦点时发生。

语法

```
public event DataGridViewCellEventHandler CellEnter
```

CellMouseClick 事件

在用户用鼠标单击单元格中的任何位置时发生。

语法

```
public event DataGridViewCellMouseEventHandler CellMouseClick
```

CellLeave 事件

在单元格失去输入焦点因而不再是当前单元格时发生。

语法

```
public event DataGridViewCellEventHandler CellLeave
```

Cells 属性

获取一个 Cells 集合，该集合表示 Bookmark 控件中的表单元格。

语法

```
[BrowsableAttribute(false)]
public Cells Cells { get; }
```

属性值：一个 Cells 集合，表示 Bookmark 控件中的表单元格。

命名空间

Microsoft.Office.Tools.Word、Microsoft.Office.Tools.Excel、System. Windows. Forms

更多资源请访问稀酷客(www.ckook.com)

CheckBox 控件

CheckBox 控件指定某个特定条件是处于打开状态还是处于关闭状态，它常用于为用户提供“是/否”或“真/假”选项。

属性

CheckBox 控件的常用属性及描述如表 C.4 所示。

表 C.4　CheckBox 控件的常用属性及描述

属　性	描　述
Appearance	此属性用于指定 CheckBox 控件的外观。可设为 Button 或 Normal
AutoCheck	使复选框单击时自动更改状态
Checked	此属性用于指定复选框是否处于选中状态，是设为 true，不是设为 false
FlatStyle	确定当用户鼠标移到控件上并单击时控件的外观
ThreeState	指定复选框是否允许使用 3 种复选状态，而不是两种，如果设为 true，则复选框的 3 种状态为“已选中”、“未选中”和“中间”

方法

CheckBox 控件的常用方法及描述如表 C.5 所示。

表 C.5　CheckBox 控件的常用方法及描述

方　法	描　述
FromHandle	返回当前与指定句柄关联的控件
Focus	该方法为控件输入焦点

事件

CheckBox 控件的常用事件及描述如表 C.6 所示。

表 C.6　CheckBox 控件的常用事件及描述

事　件	描　述
Move	在移动控件时发生
CheckedChanged	该事件在 Checked 属性值被修改时发生

Checked 属性

指定复选框是否处于选中状态。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public bool Checked { get; set; }
```

属性值：如果 CheckBox 处于选中状态，则为 true，否则为 false。默认值为 false。

注意

如果将 ThreeState 属性设置为 true，则 Checked 属性将为 Checked 或 IndeterminateCheckState 返回 true。

CheckedChanged 事件

当 Checked 属性的值更改时发生。

语法

```
public event EventHandler CheckedChanged
```

Children 属性

获取 Active Directory 层次结构中的节点的子项。

语法

```
public DirectoryEntries Children { get; }
```

属性值：一个 DirectoryEntries 对象，包含 Active Directory 层次结构中的此节点的子项。

说明

Children 属性是一个集合，它与通常的迭代功能一起，提供 Add 方法，通过该方法将节点直接添加到当前绑定到的父节点之下的集合。当向集合中添加节点时，必须指定新节点的名称，以及要与该节点相关联的架构模板的名称。

命名空间

System.DirectoryServices

更多资源请访问稀酷客(www.ckook.com)

示例

本示例使用 Children 属性的 Add 方法向 directoryEntry1 控件中添加节点。

```
    directoryEntry1.Children.Add("c:\\",directoryEntry1.SchemaClassN
ame);
```

Clear 方法

清空控件中的内容。

语法

```
    public void Clear ()
```

注意

可以使用此方法清除控件的内容，而不是给 Text 属性分配一个空字符串。

Click 事件

单击控件时触发该事件。

语法

```
    public event EventHandler Click
```

Clone 方法

创建 Bitmap 对象（它由 Rectangle 结构和指定的 PixelFormat 枚举定义）的某个部分的副本。

语法

```
    public Bitmap Clone (
        Rectangle rect,
        PixelFormat format)
```

Clone 方法的语法中各参数的说明如表 C.7 所示。

更多资源请访问稀酷客(www.ckook.com)

表 C.7　Clone 方法的语法中各参数的说明

参　数	描　述
rect	定义 Bitmap 对象中要复制的部分，坐标相对于该 Bitmap 对象
format	为目标 Bitmap 对象指定 PixelFormat 枚举
返回值	创建的新 Bitmap 图像



命名空间

System.Drawing

Close 方法

关闭对象并释放所有关联的资源。



语法

- 各种流对象（比如 StreamReader、StreamWrite、FileStream 等）的 Close 方法

```
public override void Close()
```

- SqlConnection 类的 Close 方法

```
public void Close()
```

- Socket 类的 Close 方法

```
public void Close()
public void Close(int timeout)
```

Socket.Close 方法的语法中各参数的说明如表 C.8 所示。

表 C.8　Socket.Close 方法的语法中各参数的说明

参　数	描　述
timeout	等待 timeout 秒以发送所有剩余数据，然后关闭该套接字

Color 属性

获取或设置用户选定的颜色。



语法

```
object. Color [= Color]
```

Color 属性的语法中各参数的说明如表 C.9 所示。

更多资源请访问稀酷客(www.ckook.com)

表 C.9 Color 属性的语法中各参数的说明

参　数	描　述
object	对象表达式
Color	用户选定的颜色

ColorDialog 控件

ColorDialog 控件表示一个通用对话框，该对话框显示可用的颜色，以及允许用户定义自定义颜色的控件。

属性

ColorDialog 控件的常用属性及描述如表 C.10 所示。

表 C.10 ColorDialog 控件的常用属性及描述

属　性	描　述
AllowFullOpen	获取或设置一个值，该值指示用户是否可以使用该对话框定义自定义颜色
AnyColor	获取或设置一个值，该值指示对话框是否显示基本颜色集中可用的所有颜色
Color	获取或设置用户选定的颜色
CustomColors	获取或设置对话框中显示的自定义颜色集
FullOpen	获取或设置一个值，该值表示用于创建自定义颜色的控件在对话框打开时是否可见
Options	获取初始化 ColorDialog 的值
ShowHelp	获取或设置一个值，该值指示在颜色对话框中是否显示“帮助”按钮
SolidColorOnly	获取或设置一个值，该值指示对话框是否限制用户只选择纯色

方法

ColorDialog 控件的常用方法及描述如表 C.11 所示。

表 C.11 ColorDialog 控件的常用方法及描述

方　法	描　述
RunDialog	在派生类中被重写时，指定通用对话框
ShowDialog	显示通用对话框

ColumnCount 属性

获取或设置 DataGridView 控件中显示的列数。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public int ColumnCount { get; set; }
```

属性值：DataGridView 控件中显示的列数。

示例

本示例将 DataGridView 控件的 ColumnCount 属性设置为 4，则在窗体中只显示 4 列数据。

```
dataGridView1.ColumnCount = 4;
```

本示例获取 DataGridView 控件中一共有几列数据。

```
MessageBox.Show(dataGridView1.ColumnCount.ToString());
```

Columns 属性

获取一个包含控件中所有列的集合。

语法

```
public DataGridViewColumnCollection Columns { get; }
```

属性值：一个 DataGridViewColumnCollection，包含 DataGridView 控件中的所有列。

示例

本示例通过 dataGridView1 控件的 Columns 属性设置其各个列的标题。

```
dataGridView1.Columns[0].Name = "编号";
dataGridView1.Columns[1].Name = "姓名";
dataGridView1.Columns[2].Name = "性别";
dataGridView1.Columns[3].Name = "年龄";
```

ComboBox 控件

ComboBox 控件用于在下拉组合框中显示数据，它结合了 TextBox 控件和 ListBox 控件的功能。使用此控件时，可以在组合框中输入文本项，也可以从列表中选择项。

属性

ComboBox 控件的常用属性及描述如表 C.12 所示。

更多资源请访问稀酷客(www.ckook.com)

表 C.12　ComboBox 控件的常用属性及描述

属　性	描　述
DataSource	获取或设置此 ComboBox 的数据源
DropDownStyle	获取或设置指定组合框样式的值
Enabled	获取或设置一个值，该值指示控件是否可以对用户交互做出响应
Items	获取一个对象，该对象表示该 ComboBox 中所包含项的集合
MaxDropDownItems	获取或设置要在 ComboBox 的下拉部分中显示的最大项数
Name	获取或设置控件的名称
SelectedIndex	获取或设置指定当前选定项的索引
SelectedItem	获取或设置 ComboBox 中当前选定的项
SelectedText	获取或设置 ComboBox 的可编辑部分中选定的文本
SelectedValue	获取或设置由 ValueMember 属性指定的成员属性的值
Text	获取或设置与此控件关联的文本
Visible	获取或设置一个值，该值指示是否显示该控件及其所有父控件

方法

ComboBox 控件的常用方法及描述如表 C.13 所示。

表 C.13　ComboBox 控件的常用方法及描述

方　法	描　述
SelectALL	选择 ComboBox 控件的可编辑部分的所有文本
Select	从 ComboBox 中选取指定的项
FindString	查找 ComboBox 中以指定字符串开始的第一个项
FindForm	检索控件所在的窗体

事件

ComboBox 控件的常用事件及描述如表 C.14 所示。

表 C.14　ComboBox 控件的常用事件及描述

事　件	描　述
SelectedIndexChanged	在 SelectedIndex 属性更改后发生
SelectedValueChanged	当 SelectedValue 属性更改时发生
TextChanged	在 Text 属性值更改时发生

Command 对象

对数据源执行查询、添加、删除和修改等各种操作。操作实现的方式可以使用 SQL 语句，也可以使用存储过程。

更多资源请访问稀酷客(www.ckook.com)

命名空间

System.Data.SqlClient、System.Data.Odbc、System.Data.OleDb、System.Data.OracleClient

属性

Command 对象的常用属性及描述如表 C.15 所示。

表 C.15　Command 对象的常用属性及描述

属　性	描　述
CommandText	获取或设置要对数据源执行的 Transact-SQL 语句或存储过程
CommandTimeout	获取或设置在终止执行命令的尝试并生成错误之前的等待时间
CommandType	获取或设置一个值，该值指示如何解释 CommandText 属性
Connection	获取或设置 SqlCommand 类的此实例使用的 SqlConnection 对象
Parameters	获取 SqlParameterCollection 对象
Transaction	获取或设置将在其中执行 SqlCommand 类的 SqlTransaction 对象
UpdatedRowSource	获取或设置命令结果在由 DbDataAdapter 类的“Update”方法使用时，如何应用于 DataRow 对象

方法

Command 对象的常用方法及描述如表 C.16 所示。

表 C.16　Command 对象的常用方法及描述

方　法	描　述
BeginExecuteNonQuery	启动此 SqlCommand 对象描述的 Transact-SQL 语句或存储过程的异步执行
BeginExecuteReader	启动此 SqlCommand 对象描述的 Transact-SQL 语句或存储过程的异步执行，并从服务器中检索一个或多个结果集
BeginExecuteXmlReader	启动此 SqlCommand 对象描述的 Transact-SQL 语句或存储过程的异步执行，并将结果作为 XmlReader 对象返回
CreateParameter	创建 SqlParameter 对象的新实例
EndExecuteNonQuery	完成 Transact-SQL 语句的异步执行
EndExecuteReader	完成 Transact-SQL 语句的异步执行，返回请求的 SqlDataReader 对象
EndExecuteXmlReader	完成 Transact-SQL 语句的异步执行，将请求的数据以 XML 形式返回
ExecuteNonQuery	对连接执行 Transact-SQL 语句并返回受影响的行数
ExecuteReader	将 CommandText 对象发送到 Connection 类并生成一个 SqlDataReader 对象

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
ExecuteScalar	执行查询，并返回查询所返回的结果集中第一行的第一列。忽略其他列或行
ExecuteXmlReader	将 CommandText 对象发送到 Connection 类并生成一个 XmlReader 对象
ResetCommandTimeout	将 CommandTimeout 属性重置为其默认值

CommandText 属性

获取或设置要对数据源执行的 Transact-SQL 语句或存储过程。

语法

```
public override string CommandText { get; set; }
```

属性值：要执行的 Transact-SQL 语句或存储过程，默认为空字符串。

命名空间

System.Data.SqlClient、System.Data.Odbc、System.Data.OleDb、System.Data.OracleClient

示例

本示例通过 SqlCommand 对象来执行一条 SQL 语句显示表中信息。

```
SqlCommand cmd = new SqlCommand();
cmd.CommandText="select * from TableName";
```

CommandType 属性

获取或设置一个值，该值指示如何解释 CommandText 属性。

语法

```
public override CommandType CommandType { get; set; }
```

属性值：CommandType 值（StoredProcedure 存储过程的名称、TableDirect 表的名称、Text SQL 文本命令）之一，默认为 Text。

命名空间

System.Data.SqlClient、System.Data.Odbc、System.Data.OleDb、System.

更多资源请访问稀酷客(www.ckook.com)

Compare 方法

String 类的静态方法，用于全面比较两个字符串对象。



语法

```
    public static int Compare(string strA, string strB)
    public static int Compare(string strA, string strB, bool ignorCase)
    public static int Compare(string strA, string strB, bool ignorCase,
CultureInfo)
    public static int Compare(string strA, int indexA, string strB, int
indexB, int length)
    public static int Compare(string strA, int indexA, string strB, int
indexB, int length, bool ignorCase)
    public static int Compare(string strA, int, string strB, int indexA,
int length, bool ignorCase, System.Globalization.CultureInfo culture)
```

Compare 方法的语法中各参数的说明如表 C.17 所示。

表 C.17 Compare 方法的语法中各参数的说明

参　数	描　述
strA 和 strB	待比较的两个字符串
ignorCase	指定考虑是否大小写，当取 true 时忽略大小写，取 false 时不忽略大小写
indexA 和 indexB	需要比较两个字符串中的子串时，indexA 和 indexB 为 strA 和 StrB 中子字符中的起始位置
length	待比较的字符串的最大长度
culture	字符串的区域性信息

Compare 方法的返回值如表 C.18 所示。

表 C.18 Compare 方法的返回值

参数条件	返 回 值
strA 小于 strB	负整数
strA 大于 strB	正整数
strA 等于 strB	0

CompareTo 方法

将当前字符串对象与另一个字符串对象作比较。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public int CompareTo(Object value)
public int CompareTo(string strB)
```

CompareTo 方法的语法中各参数的说明如表 C.19 所示。

表 C.19　CompareTo 方法的语法中各参数的说明

参　数	描　述
value	计算为 String 的 Object
strB	String 字符串

CompareTo 方法的返回值如表 C.20 所示。

表 C.20　CompareTo 方法的返回值

参数条件	返 回 值
strA 小于 strB	负整数
strA 大于 strB	正整数
strA 等于 strB	0

示例

本示例使用 CompareTo 方法比较两个字符串。

```
System.String StrA = "Hello";
System.String StrB = "HI";
Console.WriteLine(StrA.CompareTo(StrB));
```

本示例使用CompareTo方法将一个字符串和一个int类型的变量进行比较。

```
System.String StrA = "Hello";
int intB = 5;
Console.WriteLine(StrA.CompareTo(intB));
```

Computer 类

Computer 类提供了用于操作计算机组件（例如音频、时钟、键盘和文件系统等）的属性。

命名空间

Microsoft.VisualBasic.Devices

属性

Computer 类的常用属性及描述如表 C.21 所示。

更多资源请访问稀酷客(www.ckook.com)

表 C.21　Computer 类的常用属性及描述

属　性	描　述
Audio	获取一个对象，该对象为用于播放声音的方法提供属性
Clipboard	获取一个对象，该对象提供用于操作剪贴板的方法
Clock	获取一个对象，该对象可提供用于从系统时钟访问当前的本地时间和协调通用时间（与格林尼治标准时间相同）的属性
FileSystem	获取一个对象，该对象提供用于处理驱动器、文件和目录的属性和方法
Info	获取一个对象，该对象提供用于获取有关计算机内存、已加载的程序集、名称和操作系统的信息的属性
Keyboard	获取一个对象，该对象提供了用于访问键盘当前状态（例如，当前按下了什么键）的属性，并提供了向活动窗口发送键击的方法
Mouse	获取一个对象，该对象提供用于获取有关本地计算机中安装的鼠标的格式和配置信息的属性
Name	获取计算机名称
Network	获取一个对象，该对象提供用于与计算机所连接的网络进行交互的属性和方法
Ports	获取一个对象，该对象提供用于访问计算机的串行端口的属性和方法
Registry	获取一个对象，该对象提供用于操作注册表的属性和方法
Screen	获取表示计算机的主显示屏幕的 Screen 对象

ComputerInfo 类

ComputerInfo 类提供了用于获取与计算机的内存、已加载程序集、名称和操作系统有关信息的属性。

命名空间

Microsoft.VisualBasic.Devices

属性

ComputerInfo 类的常用属性及描述如表 C.22 所示。

表 C.22　ComputerInfo 类的常用属性及描述

属　性	描　述
AvailablePhysicalMemory	获取计算机的可用物理内存总量
AvailableVirtualMemory	获取计算机的可用虚拟地址空间的总量
InstalledUICulture	获取随操作系统安装的当前用户界面区域性

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
OSFullName	获取操作系统的全名
OSPlatform	获取计算机的操作系统的平台标识符
OSVersion	获取计算机操作系统的版本
TotalPhysicalMemory	获取计算机的物理内存总量
TotalVirtualMemory	获取计算机的可用虚拟地址空间的总量

典型应用

本实例通过使用 ComputerInfo 类的相关属性来对内存使用状态进行监控，实例运行结果如图 C.1 所示。

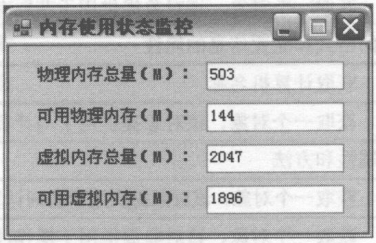


图 C.1　内存使用状态监控

实现关键代码如下：

```
    private void timer1_Tick(object sender, EventArgs e)
    {
        Computer myComputer = new Computer( );
        //获取系统的物理内存总量
        textBox1.Text = Convert.ToString(myComputer.Info.TotalPhysical
Memory / 1024 / 1024);
        //获取系统的可用物理内存
        textBox2.Text = Convert.ToString(myComputer.Info.AvailablePhysical
Memory / 1024 / 1024);
         //获取系统的虚拟内存总量
         textBox3.Text = Convert.ToString(myComputer.Info.TotalVirtual
Memory / 1024 / 1024);
         //获取系统的可用虚拟内存
         textBox4.Text = Convert.ToString(myComputer.Info.AvailableVirtual
Memory / 1024 / 1024);
               timer1.Interval = 100;
        Form1_Load(sender, e);
    }
    private void Form1_Load(object sender, EventArgs e)
```


更多资源请访问稀酷客(www.ckook.com)

```
{
    timer1.Start( );//启动计时器，以控制窗体的即时更新
}
```

Connection 对象

Connection 对象也称为连接对象，在进行任何与数据库的操作之前，首先需要完成创建与目标数据库的连接，ADO.NET 提供了与数据源连接的对象，主要包括 OleDbConnection 对象、SqlConnection 对象和 OdbcConnection 对象。

命名空间

System.Data.SqlClient、System.Data.Odbc、System.Data.OleDb、System.Data.OracleClient

ConnectionState 枚举

此枚举描述与数据源连接的当前状态。

语法

```
public enum ConnectionState
```

ConnectionState 枚举值及说明如表 C.23 所示。

表 C.23　ConnectionState 枚举值及说明

枚 举 值	描　　述
Broken	与数据源的连接中断。只有在连接打开之后才可能发生这种情况。可以关闭处于这种状态的连接，然后重新打开
Closed	连接处于关闭状态
Connecting	连接对象正在与数据源连接
Executing	连接对象正在执行命令
Fetching	连接对象正在检索数据
Open	连接处于打开状态

命名空间

System.Data.SqlClient、System.Data.Odbc、System.Data.OleDb、System.Data.OracleClient

更多资源请访问稀酷客(www.ckook.com)

ContextMenuStrip 控件

快捷菜单。

属性

ContextMenuStrip 控件的常用属性及描述如表 C.24 所示。

表 C.24 ContextMenuStrip 控件的常用属性及描述

属　性	描　述
DisplayedItems	获取当前在 ContextMenuStrip 上显示的项的子集，其中包括自动添加到 ContextMenuStrip 中的项
ImageList	获取或设置包含 ContextMenuStrip 项上显示的图像的图像列表
Items	获取属于 ContextMenuStrip 的所有项
Name	获取或设置控件的名称

continue 语句

continue 语句使执行流程跳过循环体的剩余部分而继续执行下一个循环。

示例

本示例中，计数器最初是从 1 到 10 进行计数，但通过将 continue 语句与表达式（i＜9）一起使用，跳过了 continue 与 for 循环体之间的语句。

```
static void Main(string[] args)
{
    Console.WriteLine("输出结果为：");
    for (int i = 1; i <= 10; i++)
    {
        if (i < 9)
        {
            continue;
        }
        Console.WriteLine(i);
    }
}
```

Control 类

Control 类是定义控件的基类，控件是带有可视化表示形式的组件。

更多资源请访问稀酷客(www.ckook.com)

属性

Control 类的常用属性及描述如表 C.25 所示。

表 C.25 Control 类的常用属性及描述

属　性	描　述
Anchor	使用这个属性，可以指定当控件的容器大小发生变化时，该控件如何响应
BackColor	控件的背景颜色
Bottom	设置这个属性，可以指定控件的底部距离窗口的顶部有多远。这与指定控件的高度不同
Dock	可以使控件靠在窗口的边界上
Enabled	把 Enabled 设置为 true 通常表示该控件可以接收用户的输入。把 Enabled 设置为 false 通常表示不能接收用户的输入
ForeColor	控件的前景色
Height	控件从底部到顶部的距离
Left	控件的左边界到窗口左边界的距离
Name	控件的名称。这个名称可以在代码中用于引用该控件
Parent	控件的父控件
Right	控件的右边界到窗口左边界的距离
TabIndex	获取或设置在控件的容器的控件的 Tab 键顺序
TabStop	获取或设置一个值，该值指示用户能否使用 Tab 键将焦点放到该控件上
Tag	获取或设置包含有关控件的数据的对象
Text	获取或设置与此控件关联的文本
Top	控件的顶部距离窗口顶部的距离
Visible	指定控件是否在运行期间可见
Width	控件的宽度

方法

Control 类的常用方法及描述如表 C.26 所示。

表 C.26 Control 类的常用方法及描述

方　法	描　述
Contains	检索一个值，该值指示指定控件是否为一个控件的子控件
Dispose	释放由 Control 使用的所有资源
Select	激活控件
SelectNextControl	激活下一个控件
SendToBack	将控件发送到 Z 顺序的后面

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
Show	向用户显示控件
Update	使控件重绘其工作区内的无效区域
Scale	缩放控件和任何子控件
IsKeyLocked	确定 Caps Lock、Num Lock 或 Scroll Lock 键是否有效
Hide	对用户隐藏控件
GetType	获取当前实例的类型
FromHandle	返回当前与指定句柄关联的控件
FindForm	检索控件所在的窗体
CreateControl	强制创建控件，包括创建句柄和任何子控件
Focus	为控件设置输入焦点
GetChildAtPoint	检索指定位置的子控件



事件

Control 类的常用事件及描述如表 C.27 所示。

表 C.27　Control 类的常用事件及描述

事　件	描　述
Click	在单击控件时引发。在某些情况下，这个事件也会在用户按下回车键时引发
DoubleClick	在双击控件时引发。处理某些控件上的 Click 事件，例如 Button 控件，表示永远不会调用 DoubleClick 事件
DragDrop	在完成拖放操作时引发。换言之，当一个对象被拖到控件上，然后用户释放鼠标按钮后，引发该事件
DragEnter	在被拖动的对象进入控件的边界时引发
DragLeave	在被拖动的对象移出控件的边界时引发
DragOver	在被拖动的对象放在控件上时引发
KeyDown	当控件有焦点时，按下一个键时发生该事件，这个事件总在 KeyPress 和 KeyUp 之前引发
KeyPress	当控件有焦点时，按下一个键时引发该事件，这个事件总是在 KeyDown 之后、KeyUp 之前引发。KeyDown 和 KeyPress 的区别是 KeyDown 传送被按下的键的键盘码，而 KeyPress 传送被按下的键的 char 值
KeyUp	当控件有焦点时，释放一个键时发生该事件，这个事件总是在 KeyDown 和 KeyPress 之后引发
GotFocus	在控件接收焦点时引发。不要用这个事件执行控件的有效性验证，而应使用 Validating 和 Validated

更多资源请访问稀酷客(www.ckook.com)

续表

事　件	描　述
LostFocus	在控件失去焦点时引发。不要用这个事件执行控件的有效性验证，而应使用 Validating 和 Validated
MouseDown	在鼠标指针指向一个控件，且鼠标按钮被按下时引发。这与 Click 事件不同，因为在按钮被按下之后，且未被释放之前引发 MouseDown
MouseMove	在鼠标滑过控件时引发
MouseUp	在鼠标指针位于控件上，且鼠标按钮被释放时引发
Paint	绘制控件时引发
Validaed	当控件的 CausesValidation 属性设置为 true，且该控件获得焦点时，引发该事件。它在 Validating 事件之后发生，表示有效性验证已经完成
Validating	当控件的 CausesValidation 属性设置为 true，且该控件获得焦点时，引发该事件。注意被验证有效性的控件是失去焦点的控件，而不是获得焦点的控件

Convert 类

功能

Convert 类用于将一个基本数据类型转换为另一个基本数据类型，返回与指定类型的值等效的类型；受支持的基类型是 Boolean、Char、SByte、Byte、Int16、Int32、Int64、UInt16、UInt32、UInt64、Single、Double、Decimal、DateTime 和 String。开发人员可根据不同的需要使用 Convert 类的公共方法实现不同数据类型的转换。

方法

Convert 类的常用方法及描述如表 C.28 所示。

表 C.28　Convert 类的常用方法及描述

方　法	描　述
FromBase64CharArray	将 Unicode 字符数组的子集（将二进制数据编码为 base 64 数字）转换成等效的 8 位无符号整数数组。参数指定输入数组的子集以及要转换的元素数
FromBase64String	将指定的 String（将二进制数据编码为 base 64 数字）转换成等效的 8 位无符号整数数组
GetHashCode	用做特定类型的哈希函数。GetHashCode 适合在哈希算法和数据结构（如哈希表）中使用
ToBase64CharArray	将 8 位无符号整数数组的子集转换为用 Base 64 数字编码的 Unicode 字符数组的等价子集

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
ToBase64String	将 8 位无符号整数数组的值转换为它的等效 String 表示形式（使用 base 64 数字编码）
ToBoolean	将指定的值转换为等效的布尔值
ToByte	将指定的值转换为 8 位无符号整数
ToChar	将指定的值转换为 Unicode 字符
ToDateTime	将指定的值转换为 DateTime
ToDecimal	将指定值转换为 Decimal 数字
ToDouble	将指定的值转换为双精度浮点数字
ToInt16	将指定的值转换为 16 位有符号整数
ToInt32	将指定的值转换为 32 位有符号整数
ToInt64	将指定的值转换为 64 位有符号整数
ToSByte	将指定的值转换为 8 位有符号整数
ToSingle	将指定的值转换为单精度浮点数字
ToString	将指定值转换为其等效的 String 表示形式
ToUInt16	将指定的值转换为 16 位无符号整数
ToUInt32	将指定的值转换为 32 位无符号整数
ToUInt64	将指定的值转换为 64 位无符号整数

Console 类

Console 类表示控制台应用程序的标准输入流、输出流和错误流，无法继承此类。

属性

Console 类的常用属性及描述如表 C.29 所示。

表 C.29　Console 类的常用属性及描述

属　性	描　述
BackgroundColor	获取或设置控制台的背景色
BufferHeight	获取或设置缓冲区的高度
BufferWidth	获取或设置缓冲区的宽度
CapsLock	获取一个值，该值指示 Caps Lock 键盘切换键是打开的还是关闭的
CursorLeft	获取或设置光标在缓冲区中的列位置
CursorSize	获取或设置光标在字符单元格中的高度
CursorTop	获取或设置光标在缓冲区中的行位置
CursorVisible	获取或设置一个值，用以指示光标是否可见

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
Error	获取标准错误输出流
ForegroundColor	获取或设置控制台的前景色
In	获取标准输入流
InputEncoding	获取或设置控制台用于读取输入的编码
KeyAvailable	获取一个值，该值指示按键操作在输入流中是否可用
LargestWindowHeight	根据当前字体和屏幕分辨率获取控制台窗口可能具有的最大行数
LargestWindowWidth	根据当前字体和屏幕分辨率获取控制台窗口可能具有的最大列数
NumberLock	获取一个值，该值指示 Num Lock 键盘切换键是打开的还是关闭的
Out	获取标准输出流
OutputEncoding	获取或设置控制台用于写入输出的编码
Title	获取或设置要显示在控制台标题栏中的标题
TreatControlCAsInput	获取或设置一个值，该值指示是将修改键 Control 和控制台键 C 的组合（Ctrl+C）视为普通输入，还是视为由操作系统处理的中断
WindowHeight	获取或设置控制台窗口区域的高度
WindowLeft	获取或设置控制台窗口区域的最左边相对于屏幕缓冲区的位置
WindowTop	获取或设置控制台窗口区域的最顶部相对于屏幕缓冲区的位置
WindowWidth	获取或设置控制台窗口的宽度



方法

Console 类的常用方法及描述如表 C.30 所示。

表 C.30　Console 类的常用方法及描述

方　法	描　述
Beep	通过控制台扬声器播放提示音
Clear	清除控制台缓冲区和相应的控制台窗口的显示信息
MoveBufferArea	将屏幕缓冲区的指定源区域复制到指定的目标区域
OpenStandardError	获取标准错误流
OpenStandardInput	获取标准输入流
OpenStandardOutput	获取标准输出流
Read	从标准输入流读取下一个字符
ReadKey	获取用户按下的下一个字符或功能键
ReadLine	从标准输入流读取下一行字符
ResetColor	将控制台的前景色和背景色设置为默认值
SetBufferSize	将屏幕缓冲区的高度和宽度设置为指定值
SetCursorPosition	设置光标位置
SetError	将 Error 属性设置为指定的 TextWriter 对象

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
SetIn	将 In 属性设置为指定的 TextReader 对象
SetOut	将 Out 属性设置为指定的 TextWriter 对象
SetWindowPosition	设置控制台窗口相对于屏幕缓冲区的位置
SetWindowSize	将控制台窗口的高度和宽度设置为指定值
Write	将指定值的文本表示形式写入标准输出流
WriteLine	将指定的数据（后跟当前行终止符）写入标准输出流

Copy 方法

将现有文件复制到新文件，允许改写同名的文件。



语法

```
public static void Copy (string sourceFileName,string destFileName)
public static void Copy (string sourceFileName,string
destFileName,bool overwrite)
```

Copy 方法的语法中各参数的说明如表 C.31 所示。

表 C.31　Copy 方法的语法中各参数的说明

参　数	描　述
sourceFileName	要复制的文件
destFileName	目标文件的名称，不能是目录
overwrite	如果可以改写目标文件，则为 True；否则为 False



命名空间

System.IO

CopyFile 方法

将文件复制到新的位置。



语法

```
public static void CopyFile (
    string sourceFileName,
    string destinationFileName)
```


更多资源请访问稀酷客(www.ckook.com)

CopyFile 方法的语法中各参数的说明如表 C.32 所示。

表 C.32　CopyFile 方法的语法中各参数的说明

参　数	描　述
sourceFileName	String 类型，要复制的文件，必选
destinationFileName	String 类型，文件应复制到的位置，必选

命名空间

Microsoft.VisualBasic.FileIO

COUNT 函数

返回组中的项的数量。

语法

```
COUNT ( { [ ALL | DISTINCT ] expression } | * ) )
```

Count 函数的语法中各参数的说明如表 C.33 所示。

表 C.33　Count 函数的语法中各参数的说明

参　数	描　述
ALL	对所有值应用聚合函数，默认值为 ALL
DISTINCT	指定 COUNT 返回唯一非空值的数量
expression	一个表达式，其类型是除 uniqueidentifier、text、image 或 ntext 之外的任何类型。不允许使用聚合函数和子查询
返回值	int 类型，组中项的数量

Count 属性

语法

- ArrayList 类的 Count 属性

获取 ArrayList 中实际包含的元素数。

```
public virtual int Count { get; }
```

属性值：ArrayList 中实际包含的元素数。

- DataGridView 控件的 Count 属性

获取集合中的行数。

```
public int Count { get; }
```


更多资源请访问稀酷客(www.ckook.com)

属性值：DataGridViewRowCollection 中的行数。

- Hashtable 类的 Count 属性

获取包含在 Hashtable 中的键/值对的数目。

```
public virtual int Count { get; }
```

属性值：包含在 Hashtable 中的键/值对的数目。

Create 方法

在指定路径中创建文件。

语法

```
public static FileStream Create (string path)
public static FileStream Create (string path,int bufferSize)
public static FileStream Create (string path,int
bufferSize,FileOptions options)
public static FileStream Create (string path,int bufferSize,
FileOptions options,FileSecurity fileSecurity)
```

Create 方法的语法中各参数的说明如表 C.34 所示。

表 C.34 Create 方法的语法中各参数的说明

参　数	描　述
path	要创建的文件的路径及名称
bufferSize	用于读取和写入文件的已放入缓冲区的字节数
options	FileOptions 值之一，它描述如何创建或改写该文件
fileSecurity	FileSecurity 值之一，它确定文件的访问控制和审核安全性
返回值	一个 FileStream，它提供对 path 中指定的文件的读/写访问

命名空间

System.IO

CreateDirectory 方法

创建指定路径中的所有目录。

语法

```
public static DirectoryInfo CreateDirectory (string path)
```


更多资源请访问稀酷客(www.ckook.com)

```
public static DirectoryInfo CreateDirectory (string
path,DirectorySecurity directorySecurity)
```

CreateDirectory 方法的语法中各参数的说明如表 C.35 所示。

表 C.35　CreateDirectory 方法的语法中各参数的说明

参　数	描　述
path	要创建的目录路径
directorySecurity	要应用于此目录的访问控制
返回值	第一种重载形式的返回值为由 path 指定的 DirectoryInfo；第二种重载形式的返回值为新创建的目录的 DirectoryInfo 对象

命名空间

System.IO

CreateSubKey 方法

创建一个新子项或打开一个现有子项以进行写访问。

语法

```
public RegistryKey CreateSubKey (string subkey)
```

CreateSubKey 方法的语法中各参数的说明如表 C.36 所示。

表 C.36　CreateSubKey 方法的语法中各参数的说明

参　数	描　述
subkey	要创建或打开的子项的名称或路径
返回值	RegistryKey 对象，表示新建的子项或空引用。如果为 subkey 指定了零长度字符串，则返回当前的 RegistryKey 对象

命名空间

Microsoft.Win32

CreateText 方法

创建或打开一个文件用于写入 UTF-8 编码的文本。

语法

```
public static StreamWriter CreateText(string path)
```


更多资源请访问稀酷客(www.ckook.com)

CompareTo 方法的语法中各参数的说明如表 C.37 所示。

表 C.37　CompareTo 方法的语法中各参数的说明

参　数	描　述
path	要打开以进行写入的文件
返回值	一个 StreamWriter，它使用 UTF-8 编码写入指定的文件

命名空间

System.IO

CryptoStream 类

定义将数据流链接到加密转换的流。

命名空间

System.Security.Cryptography

构造函数

```
public CryptoStream(
    Stream stream,
    ICryptoTransform transform,
    CryptoStreamMode mode)
```

CryptoStream 类的构造函数中各参数的说明如表 C.38 所示。

表 C.38　CryptoStream 类的构造函数的语法中各参数的说明

参　数	描　述
stream	对其执行加密转换的流
transform	要对流执行的加密转换
mode	CryptoStreamMode 枚举值之一

属性

CryptoStream 类的常用属性及描述如表 C.39 所示。

表 C.39　CryptoStream 类的常用属性及描述

属　性	描　述
CanRead	获取一个值，该值指示当前的 CryptoStream 是否可读
CanSeek	获取一个值，该值指示您是否可以在当前 CryptoStream 中搜索
CanTimeout	获取一个值，该值确定当前流是否可以超时

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
CanWrite	获取一个值，该值指示当前的 CryptoStream 是否可写
Length	获取用字节表示的流长度
Position	获取或设置当前流中的位置
ReadTimeout	获取或设置一个值（以毫秒为单位），该值确定流在超时前尝试读取多长时间
WriteTimeout	获取或设置一个值（以毫秒为单位），该值确定流在超时前尝试写入多长时间



方法

CryptoStream 类的常用方法及描述如表 C.40 所示。

表 C.40　CryptoStream 类的常用方法及描述

方　法	描　述
BeginRead	开始异步读操作
BeginWrite	开始异步写操作
Clear	释放由 CryptoStream 使用的所有资源
Close	关闭当前流并释放与之关联的所有资源（如套接字和文件句柄）
EndRead	等待挂起的异步读取完成
EndWrite	结束异步写操作
Flush	清除该流的所有缓冲区，使得所有缓冲的数据都被写入到基础设备
Read	从当前 CryptoStream 中读取字节序列，并将流中的位置提升读取的字节数
ReadByte	从流中读取一个字节，并将流内的位置向前推进一个字节，或者如果已到达流的末尾，则返回-1
Seek	设置当前流中的位置
SetLength	设置当前流的长度
Write	将一个字节序列写入当前 CryptoStream，并将流中的当前位置提升写入的字节数
WriteByte	将一个字节写入流内的当前位置，并将流内的位置向前推进一个字节

CrystalReportViewer 控件

CrystalReportViewer 控件允许在应用程序中查看 Crystal 报表，其 ReportSource 属性用于设置要查看哪个报表。该属性设置之后，报表将显示在 CrystalReportViewer 控件中，报表源可以是 ReportDocument 和报表文件的路径，也可以是强类型报表。

更多资源请访问稀酷客(www.ckook.com)

属性

CrystalReportViewer 控件的常用属性及描述如表 C.41 所示。

表 C.41 CrystalReportViewer 控件的常用属性及描述

属 性	描 述
DasplayGroupTree	获取或设置树视图是可见的还是隐藏的
DisplayStatusBar	获取或设置状态栏是可见的还是隐藏的
DisplayToolbar	获取或设置工具栏是可见的还是隐藏的
EnableToolTips	获取或设置是否在报表查看器中显示工具提示
ParameterFidldInfo	获取或设置参数字段集合
ReportSource	将报表设置为绑定到 CrystalReportViewer 控件
SelectionFormula	获取或设置报表的记录选定公式
ShowCloseButton	获取或设置 CrystalReportViewer 控件工具栏是否有用于关闭报表页面的按钮
ShowExportButton	获取或设置 CrystalReportViewer 控件工具栏是否有用于导出的按钮
ShowGotoPageButton	获取或设置 CrystalReportViewer 控件工具栏是否有用于指定页码的按钮
ShowGroupTreeButton	获取或设置 CrystalReportViewer 控件工具栏是否有用于显示或隐藏组树的按钮
ShowPageNavigateButtons	获取或设置 CrystalReportViewer 控件工具栏是否有用于页面导航的按钮
ShowPrintButton	获取或设置工具栏上的打印按钮是可见的还是隐藏的
ShowRefreshButton	获取或设置工具栏上的刷新按钮是可见的还是隐藏的
ShowTextSearchButton	获取或设置工具栏上的搜索按钮是可见的还是隐藏的
ShowZoomButton	获取或设置工具栏上的缩放因数列表是可见的还是隐藏的
ViewTimeSelectionFormula	获取或设置记录选定公式以筛选报表当前使用的数据

方法

CrystalReportViewer 控件的常用方法及描述如表 C.42 所示。

表 C.42 CrystalReportViewer 控件的常用方法及描述

方 法	描 述
CloseView	关闭 CrystalReportViewer 控件中的视图选项卡
DrillDownOnGroup	在组上向下钻取
ExportReport	导出 CrystalReportViewer 控件中显示的报表
GetCurrentPageNumber	获取报表的当前页码
PrintReport	打印 CrystalReportViewer 控件中显示的报表
RefreshReport	刷新 CrystalReportViewer 控件中显示的报表

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
SearchForText	搜索报表以查找给定文本
ShowFirstPage	显示报表的第一页
ShowGroupTree	在 CrystalReportViewer 控件中显示组树
ShowLastPage	显示报表的最后一页
ShowNextPage	显示报表的下一页
ShowNthPage	显示报表的指定页
ShowPreviousPage	显示报表的上一页
Zoom	更改 CrystalReportViewer 控件的放大倍数

事件

CrystalReportViewer 控件的常用事件及描述如表 C.43 所示。

表 C.43　CrystalReportViewer 控件的常用事件及描述

事　件	描　述
Drill	当向下钻取报表时事件激活
DrillDownSubreport	当用户向下钻取子报表时事件激活
Navigate	当用户浏览报表时事件激活
ReportRefresh	当刷新报表中的数据时事件激活
Search	在报表中搜索文本时事件激活
ViewZoom	当更改 CrystalReportViewer 控件的缩放比例时事件激活

CurrentCell 属性

获取或设置当前处于活动状态的单元格。

语法

```
public DataGridViewCell CurrentCell { get; set; }
```

属性值：当前单元格的 DataGridViewCell，如果没有当前单元格，则为 NULL，默认值是第一列中的第一个单元格。

CurrentRow 属性

获取包含当前单元格的行。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public DataGridViewRow CurrentRow { get; }
```

属性值：当前单元格所在的行的 DataGridViewRow，如果没有当前单元格，则为 NULL。

Cursor 类

代表用于绘制鼠标指针的图像。

属性

Cursor 类的常用属性及描述如表 C.44 所示。

表 C.44　Cursor 类的常用属性及描述

属　性	描　述
Clip	获取或设置表示光标的剪辑矩形的边界
Current	获取或设置代表鼠标光标的光标对象
Handle	获取光标句柄
HotSpot	获取光标作用点
Position	获取或设置光标位置
Size	获取光标对象的大小
Tag	获取或设置包含有关 Cursor 的数据的对象

方法

Cursor 类的常用方法及描述如表 C.45 所示。

表 C.45　Cursor 类的常用方法及描述

方　法	描　述
CopyHandle	复制 Cursor 的句柄
Draw	在指定边界内、指定的表面上绘制光标
DrawStretched	在指定边界内、指定的表面上以拉伸格式绘制光标
Hide	隐藏光标
Show	显示光标

Cut 方法

将文本框中的当前选定内容移动到“剪贴板”中。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public void Cut ()
```

示例

本示例使用 Cut 方法将 textBox1 文本框中的选定内容添加到“剪贴板”上。

```
if(textBox1.SelectedText != "")
    textBox1.Cut();
```

本示例使用 Cut 方法将 richtextBox1 中的选定内容添加到“剪贴板”上。

```
if(richtextBox1.SelectedText != "")
    richtextBox1.Cut();
```

常量

常量又叫常数，是在程序运行过程中其值不改变的量。在 C#中，常量的数据类型有多种，分别是：sbyte、byte、short、ushort、int、uint、long、ulong、char、float、double、decimal、bool、string 等。

常量被声明为字段，通过 const 关键字声明，常量必须在声明时初始化。例如：

```
public const int months = 12;
```

可以同时声明多个相同类型的常量，例如：

```
const int months = 12, weeks = 52, days = 365;
```



注意

不允许在常量声明中使用 static 修饰符。

更多资源请访问稀酷客(www.ckook.com)

D

DataAdapter 类

功能

DataAdapter 类表示一组 SQL 命令和一个数据库连接，它们用于填充 DataSet 和更新数据源。

DataAdapter 类用做 DataSet 和数据源之间的桥接器以便检索和保存数据，它通过映射 Fill 和 Update 方法来提供这一桥接器。

如果所连接的是 SQL Server 数据库，则可以通过将 SqlDataAdapter 与关联的 SqlCommand 和 SqlConnection 对象一起使用，从而提高总体性能。对于支持 OLE DB 的数据源，请使用 DataAdapter 及其关联的 OleDbCommand 和 OleDbConnection 对象。对于支持 ODBC 的数据源，请使用 DataAdapter 及其关联的 OdbcCommand 和 OdbcConnection 对象。对于 Oracle 数据库，请使用 DataAdapter 及其关联的 OracleCommand 和 OracleConnection 对象。

命名空间

System.Data.SqlClient、System.Data.Odbc、System.Data.OleDb、System.Data.OracleClient

属性

DataAdapter 类的常用属性及描述如表 D.1 所示。

表 D.1　DataAdapter 类的常用属性及描述

属　性	描　述
AcceptChangesDuringFill	获取或设置一个值，该值指示在任何 Fill 操作过程中，在将 AcceptChanges 添加到 DataTable 之后是否在 DataRow 上调用它
AcceptChangesDuringUpdate	获取或设置在 Update 期间是否调用 AcceptChanges
FillLoadOption	获取或设置 LoadOption，后者确定适配器如何从 DbDataReader 中填充 DataTable

方法

DataAdapter 类的常用方法及描述如表 D.2 所示。

更多资源请访问稀酷客(www.ckook.com)

表 D.2　DataAdapter 类的常用方法及描述

方　法	描　述
Fill	在 DataSet 中添加或刷新行以与数据源中的行匹配
FillSchema	向指定的 DataSet 添加一个 DataTable
GetFillParameters	获取当执行 SQL SELECT 语句时由用户设置的参数
Update	从名为“Table”的 DataTable 为指定的 DataSet 中每个已插入、已更新或已删除的行调用相应的 INSERT、UPDATE 或 DELETE 语句

DataGridView 控件

DataGridView 控件提供一种强大而灵活的以表格形式显示数据的方式。可以使用 DataGridView 控件来显示少量数据的只读视图，也可以对其进行缩放以显示特大数据集的可编辑视图。



属性

DataGridView 控件的常用属性及描述如表 D.3 所示。

表 D.3　DataGridView 控件的常用属性及描述

属　性	描　述
AllowUserToAddRows	获取或设置一个值，该值指示是否向用户显示添加行的选项
AllowUserToDeleteRows	获取或设置一个值，该值指示是否允许用户从 DataGridView 中删除行
AllowUserToResizeColumns	获取或设置一个值，该值指示用户是否可以调整列的大小
AllowUserToResizeRows	获取或设置一个值，该值指示用户是否可以调整行的大小
ColumnCount	获取或设置 DataGridView 中显示的列数
ColumnHeadersVisible	获取或设置一个值，该值指示是否显示列标题行
Columns	获取一个包含控件中所有列的集合
CurrentCell	获取或设置当前处于活动状态的单元格
CurrentRow	获取包含当前单元格的行
DataMember	获取或设置在 DataGridView 控件上显示的数据源 DataSource 中的列表或表的名称
DataSource	获取或设置在 DataGridView 控件上显示数据的数据源
Enabled	获取或设置一个值，该值指示控件是否可以对用户交互做出响应
Font	获取或设置 DataGridView 显示的文本字体
ForeColor	获取或设置 DataGridView 的前景色
GridColor	获取和设置网格线的颜色，网格线对 DataGridView 的单元格进行分隔
MultiSelect	获取或设置一个值，该值指示是否允许用户一次选择 DataGridView 的多个单元格、行或列

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
Name	获取或设置控件的名称
NewRowIndex	获取新记录所在行的索引
ReadOnly	获取一个值，该值指示用户是否可以编辑 DataGridView 控件的单元格
RowCount	获取或设置 DataGridView 中显示的行数
Rows	获取一个集合，该集合包含 DataGridView 控件中的所有行
SelectedCells	获取用户选定的单元格的集合
SelectedColumns	获取用户选定的列的集合
SelectedRows	获取用户选定的行的集合
SortedColumn	获取 DataGridView 内容的当前排序所依据的列
SortOrder	获取一个值，该值指示是按升序或降序对 DataGridView 控件中的项进行排序，还是不排序



方法

DataGridView 控件的常用方法及描述如表 D.4 所示。

表 D.4　DataGridView 控件的常用方法及描述

方　法	描　述
AutoResizeColumns	调整所有列的宽度以适应其单元格的内容
AutoResizeRowHeadersWidth	调整行标题的宽度以适应标题内容
AutoResizeRows	调整某些或所有行的高度以适应其内容
BeginEdit	将当前的单元格置于编辑模式下
CancelEdit	取消当前选定单元格的编辑模式并丢弃所有更改
CommitEdit	将当前单元格中的更改提交到数据缓存，但不结束编辑模式
DisplayedColumnCount	返回向用户显示的列数
DisplayedRowCount	返回向用户显示的行数
EndEdit	提交对当前单元格进行的编辑并结束编辑操作
GetCellCount	获取满足所提供筛选器的单元格的数目
SelectAll	选择 DataGridView 中的所有单元格
Sort	对 DataGridView 控件的内容进行排序



示例

DataGridView 控件的常用事件及描述如表 D.5 所示。

更多资源请访问稀酷客(www.ckook.com)

表 D.5　DataGridView 控件的常用事件及描述

事　件	描　述
CancelRowEdit	在 DataGridView 控件的 VirtualMode 属性为 true 并且取消行中的编辑时发生
CellClick	在单元格的任何部分被单击时发生
CellDoubleClick	在用户双击单元格中的任何位置时发生
CellEndEdit	在为当前选定的单元格停止编辑模式时发生
CellEnter	在 DataGridView 控件中的当前单元格更改或者该控件接收到输入焦点时发生
CellLeave	在单元格失去输入焦点因而不再是当前单元格时发生
CellMouseClick	在用户用鼠标单击单元格中的任何位置时发生
CellMouseDoubleClick	在双击 DataGridView 中的单元格时发生
CellMouseDown	当鼠标指针位于单元格的边界内时，在用户按鼠标按钮时发生
CellMouseEnter	在鼠标指针进入单元格时发生
CellMouseLeave	在鼠标指针离开单元格时发生
CellMouseMove	在鼠标指针移到 DataGridView 控件上时发生
CellMouseUp	当鼠标指针停留在单元格上时，在用户松开鼠标按钮时发生
CellValueChanged	在单元格的值更改时发生
ColumnAdded	在向控件添加一列时发生
ColumnHeaderCellChanged	在列标题单元格的内容更改时发生
ColumnHeaderMouseClick	在用户单击列标题时发生
ColumnHeaderMouseDoubleClick	在双击列标题时发生
ColumnRemoved	在从控件中移除列时发生
ColumnSortModeChanged	当列的 SortMode 属性值更改时发生
CurrentCellChanged	当 CurrentCell 属性更改时发生
RowsAdded	在向 DataGridView 中添加新行之后发生
RowsRemoved	在从 DataGridView 中删除一行或多行时发生
SelectionChanged	在当前选择出现更改时发生
Sorted	在 DataGridView 控件完成排序操作时发生

DataMember 属性

获取或设置在 DataGridView 控件上显示的数据源 DataSource 中的列表或表的名称。

语法

```
public string DataMember { get; set; }
```


更多资源请访问稀酷客(www.ckook.com)

属性值：DataSource 中 DataGridView 显示其数据的表或列表的名称。默认值为 Empty。

DataReader 类

功能

DataReader 类是一个简单的数据集，用于从数据源中检索只读数据集，常用于检索大量数据。DataReader 类只允许以只读、顺向的方式查看其中所存储的数据，提供一个非常有效率的数据查看模式，同时它还是一种非常节省资源的数据对象。

注意

每次打开一个新对象，必须关闭前一个 DataReader 对象，因为它是以独占方式与数据库交互。否则，会接收到其产生的异常。

命名空间

System.Data.SqlClient、System.Data.Odbc、System.Data.OleDb、System.Data.OracleClient

属性

DataReader 类的常用属性及描述如表 D.6 所示。

表 D.6　DataReader 类的常用属性及描述

属　性	描　述
FieldCount	获取当前行的列数
RecordsAffected	获取执行 SQL 语句所更改、添加或删除的行数

方法

DataReader 类的常用方法及描述如表 D.7 所示。

表 D.7　DataReader 类的常用方法及描述

方　法	描　述
Read	使 DataReader 对象前进到下一条记录
Close	关闭 DataReader 对象
Get	用来读取数据集的当前行的某一列的数据

更多资源请访问稀酷客(www.ckook.com)

DataSet 类

功能

DataSet 类是支持 ADO.NET 的断开式、分布式数据方案的核心对象，它是数据的内存驻留表示形式，无论数据源是什么，它都会提供一致的关系编程模型，另外，它还可以用于多种不同的数据源，比如用于 XML 数据或用于管理应用程序本地的数据。DataSet 类表示包括相关表、约束和表间关系在内的整个数据集。

命名空间

System.Data

属性

DataSet 类的常用属性及描述如表 D.8 所示。

表 D.8 DataSet 类的常用属性及描述

属　性	描　述
CaseSensitive	获取或设置一个值，该值指示 DataTable 对象中的字符串比较是否区分大小写
DataSetName	获取或设置当前 DataSet 的名称
HasErrors	获取一个值，指示在此 DataSet 中的任何 DataTable 对象中是否存在错误
Namespace	获取或设置 DataSet 的命名空间
Relations	获取用于将表链接起来并允许从父表浏览到子表的关系的集合
Tables	获取包含在 DataSet 中的表的集合

方法

DataSet 类的常用方法及描述如表 D.9 所示。

表 D.9 DataSet 类的常用方法及描述

方　法	描　述
AcceptChanges	提交自加载此 DataSet 或上次调用 AcceptChanges 以来对其进行的所有更改
Clear	通过移除所有表中的所有行来清除任何数据的 DataSet
Copy	复制该 DataSet 的结构和数据
CreateDataReader	为每个 DataTable 返回带有一个结果集的 DataTableReader，顺序与 Tables 集合中表的显示顺序相同

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
GetXml	返回存储在 DataSet 中的数据的 XML 表示形式
GetXmlSchema	返回存储在 DataSet 中的数据的 XML 表示形式的 XML 架构
HasChanges	获取一个值，该值指示 DataSet 是否有更改，包括新增行、已删除的行或已修改的行
Load	通过所提供的 IDataReader，用某个数据源的值填充 DataSet
Merge	将指定的 DataSet、DataTable 或 DataRow 对象的数组合并到当前的 DataSet 或 DataTable 中
ReadXml	将 XML 架构和数据读入 DataSet
ReadXmlSchema	将 XML 架构读入 DataSet
Reset	将 DataSet 重置为其初始状态
WriteXml	从 DataSet 写 XML 数据，还可以选择写架构
WriteXmlSchema	写 XML 架构形式的 DataSet 结构

DataSource 属性

用于获取或设置 DataGridView 控件所显示数据的数据源。

语法

```
public Object DataSource { get; set; }
```

属性值：包含 DataGridView 控件要显示的数据的对象。

DATEADD 函数

在向指定日期加上一段时间的基础上，返回新的 datetime 值。

语法

```
DATEADD ( datepart , number, date )
```

DATEADD 函数的语法中各参数的说明如表 D.10 所示。

表 D.10　DATEADD 函数的语法中各参数的说明

参　数	描　述
datepart	规定应向日期的哪一部分返回新值的参数
number	用来增加 datepart 的值。如果指定一个不是整数的值，则将废弃此值的小数部分
date	返回 datetime 或 smalldatetime 值或日期格式字符串的表达式
返回值	返回 datetime，但如果 date 参数是 smalldatetime，则返回 smalldatetime

更多资源请访问稀酷客(www.ckook.com)

示例

本示例在订单表中将所有订单到期日期都推迟 3 天，并使用以下语句得到新的日期。

```
SELECT DATEADD(day, 3, OrderDate)
FROM tb_Order
```

本示例在订单表中将所有订单到期日期都推迟 3 月，并使用以下语句得到新的日期。

```
SELECT DATEADD(month, 3, OrderDate)
FROM tb_Order
```

本示例在订单表中将所有订单到期日期都推迟 3 年，并使用以下语句得到新的日期。

```
SELECT DATEADD(year, 3, OrderDate)
FROM tb_Order
```

DateDiff 方法

获取日期时间的间隔数。

语法

```
public static long DateDiff (
    DateInterval/ string Interval,
    DateTime Date1,
    DateTime Date2,
    [OptionalAttribute] FirstDayOfWeek DayOfWeek,
    [OptionalAttribute] FirstWeekOfYear WeekOfYear)
```

DateDiff 方法的语法中各参数的说明如表 D.11 所示。

表 D.11　DateDiff 方法的语法中各参数的说明

参　数	描　述
Interval	DateInterval 枚举值或 String 表达式，表示要用做 Date1 和 Date2 之差的单位的时间间隔
Date1	要在计算中使用的第一个日期/时间值
Date2	要在计算中使用的第二个日期/时间值
DayOfWeek	从 FirstDayOfWeek 枚举中选择的值，用于指定一周的第一天。如果未指定，则使用 FirstDayOfWeek.Sunday
WeekOfYear	从 FirstWeekOfYear 枚举中选择的值，用于指定一年的第一周。如果未指定，则使用 FirstWeekOfYear.Jan1
返回值	返回一个 Long 值，用于指定两个 Date 值之间的时间间隔数

更多资源请访问稀酷客(www.ckook.com)

命名空间

Microsoft.VisualBasic

DateTime 结构

DateTime 结构用来表示时间上的一刻，通常以日期和当天的时间表示。

属性

DateTime 结构的常用属性及描述如表 D.12 所示。

表 D.12　DateTime 结构的常用属性及描述

属　性	描　述
Date	获取此实例的日期部分
Day	获取此实例所表示的日期为该月中的第几天
DayOfWeek	获取此实例所表示的日期是星期几
DayOfYear	获取此实例所表示的日期是该年中的第几天
Hour	获取此实例所表示日期的小时部分
Millisecond	获取此实例所表示日期的毫秒部分
Minute	获取此实例所表示日期的分钟部分
Month	获取此实例所表示日期的月份部分
Now	获取一个 DateTime 对象，该对象设置为此计算机上的当前日期和时间，表示为本地时间
Second	获取此实例所表示日期的秒部分
Ticks	获取表示此实例的日期和时间的刻度数
TimeOfDay	获取此实例的当天的时间
Today	获取当前日期
Year	获取此实例所表示日期的年份部分

方法

DateTime 结构的常用方法及描述如表 D.13 所示。

表 D.13　DateTime 结构的常用方法及描述

方　法	描　述
Add	将指定的 TimeSpan 的值加到此实例的值上
AddDays	将指定的天数加到此实例的值上
AddHours	将指定的小时数加到此实例的值上
AddMilliseconds	将指定的毫秒数加到此实例的值上

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
AddMinutes	将指定的分钟数加到此实例的值上
AddMonths	将指定的月份数加到此实例的值上
AddSeconds	将指定的秒数加到此实例的值上
AddTicks	将指定的刻度数加到此实例的值上
AddYears	将指定的年份数加到此实例的值上
Compare	比较 DateTime 的两个实例，并返回它们相对值的指示
DaysInMonth	返回指定年和月中的天数
FromBinary	反序列化一个 64 位二进制值，并重新创建序列化的 DateTime 初始对象
FromFileTime	将指定的 Windows 文件时间转换为等效的本地时间
IsLeapYear	返回指定的年份是否为闰年的指示
Parse	将日期和时间的指定字符串表示形式转换为其等效的 DateTime
ParseExact	将日期和时间的指定字符串表示形式转换为其等效的 DateTime。字符串表示形式的格式必须与指定的格式完全匹配
Subtract	从此实例中减去指定的时间或持续时间
ToBinary	将当前 DateTime 对象序列化为一个 64 位二进制值，该值随后可用于重新创建 DateTime 对象
ToFileTime	将当前 DateTime 对象的值转换为 Windows 文件时间
ToLocalTime	将当前 DateTime 对象的值转换为本地时间
ToLongDateString	将当前 DateTime 对象的值转换为其等效的长日期字符串表示形式
ToLongTimeString	将当前 DateTime 对象的值转换为其等效的长时间字符串表示形式
ToShortDateString	将当前 DateTime 对象的值转换为其等效的短日期字符串表示形式
ToShortTimeString	将当前 DateTime 对象的值转换为其等效的短时间字符串表示形式
TryParse	将日期和时间的指定字符串表示形式转换为其等效的 DateTime
TryParseExact	将日期和时间的指定字符串表示形式转换为其等效的 DateTime。字符串表示形式的格式必须与指定的格式完全匹配

DateTimePicker 控件

使用 DateTimePicker 控件，用户可以从日期或时间列表中选择单个项，用来表示日期，该控件显示为两部分：一部分为下拉列表（以文本形式表示的日期），另一部分为网格（在单击列表旁边的向下箭头时显示）。

属性

DateTimePicker 控件的常用属性及描述如表 D.14 所示。

更多资源请访问稀酷客(www.ckook.com)

表 D.14　DateTimePicker 控件的常用属性及描述

属　性	描　述
CalendarFont	应用于日历的字体样式
CalendarForeColor	日历的前景色
CalendarTitleForeColor	日历标题的前景色
CalendarTrailingForeColor	日历结尾日期的前景色
Checked	是否已用有效日期/时间值设置了 Value 属性且显示的值可以更新
MaxDate	确定选择日期和时间的最大值
MinDate	确定选择日期和时间的最小值
ShowUpDown	是否修改控件值以显示数字，而不是下拉日历
Value	当前控件的日期或时间值

方法

DateTimePicker 控件的常用方法及描述如表 D.15 所示。

表 D.15　DateTimePicker 控件的常用方法及描述

方　法	描　述
ToString	返回表示当前 DateTimePicker 控件的字符串

事件

DateTimePicker 控件的常用事件及描述如表 D.16 所示。

表 D.16　DateTimePicker 控件的常用事件及描述

事　件	描　述
Click	在单击控件时发生
CloseUp	当下拉日历被关闭并消失时发生
ValueChanged	当 Value 属性更改时发生

DAY 函数

返回代表指定日期的天的日期部分的整数。

语法

```
DAY ( date )
```

DAY 函数的语法中各参数的说明如表 D.17 所示。

更多资源请访问稀酷客(www.ckook.com)

表 D.17　DAY 函数的语法中各参数的说明

参　数	描　述
date	类型为 datetime 或 smalldatetime 的表达式
返回值	int，表示日期中的当前天数

示例

本示例获取指定日期中的天数，返回结果为 23。

```
SELECT DAY('11/23/2006')
```

DayOfWeek 属性

获取此实例所表示的日期是星期几。

语法

```
public DayOfWeek DayOfWeek { get; }
```

属性值：一个 DayOfWeek 枚举常数，它指示星期几，该属性值的范围从零（表示星期日）到六（表示星期六）。

DayOfWeek 枚举值的说明如表 D.18 所示。

表 D.18　DayOfWeek 枚举值的说明

枚 举 值	描　述
Sunday	表示星期日
Monday	表示星期一
Tuesday	表示星期二
Wednesday	表示星期三
Thursday	表示星期四
Friday	表示星期五
Saturday	表示星期六

Delete 方法

语法

- File 类的 Delete 方法

删除指定的文件。如果指定的文件不存在，则不引发异常。

```
public static void Delete(string path)
```


更多资源请访问稀酷客(www.ckook.com)

File 类的 Delete 方法的语法中各参数的说明如表 D.19 所示。

表 D.19　File 类的 Delete 方法的语法中各参数的说明

参　数	描　述
path	要删除的文件的名称

- FileInfo 类的 Delete 方法

永久删除文件。

```
public override void Delete()
```

- Directory 类的 Delete 方法

从指定路径删除空目录。

```
public static void Delete(string path)
public static void Delete(string path,bool recursive)
```

Directory 类的 Delete 方法的语法中各参数的说明如表 D.20 所示。

表 D.20　Directory 类的 Delete 方法的语法中各参数的说明

参　数	描　述
path	要移除的空目录的名称。此目录必须为可写或为空
recursive	若要移除 path 中的目录、子目录和文件，则为 true；否则为 false

- DirectoryInfo 类的 Delete 方法

从路径中删除 DirectoryInfo 及其内容。

```
public override void Delete()
public void Delete(bool recursive)
```

DirectoryInfo 类的 Delete 方法的语法中各参数的说明如表 D.21 所示。

表 D.21　DirectoryInfo 类的 Delete 方法的语法中各参数的说明

参　数	描　述
recursive	若为 true，则删除此目录、其子目录以及所有文件；否则为 false



命名空间

System.IO

Directory 类

Directory 类公开了用于创建、移动、枚举、删除目录和子目录的静态方法。

更多资源请访问稀酷客(www.ckook.com)



说明

由于所有的 Directory 方法都是静态的，所以如果只想执行一个操作，那么使用 Directory 类中方法的效率比使用相应的 DirectoryInfo 实例方法可能更高。

命名空间

System.IO

方法

Directory 类的常用方法及描述如表 D.22 所示。

表 D.22　Directory 类的常用方法及描述

方　法	描　述
CreateDirectory	创建指定路径中的所有目录
Delete	删除指定的目录
Exists	确定给定路径是否引用磁盘上的现有目录
GetCreationTime	获取目录的创建日期和时间
GetDirectories	获取指定目录中子目录的名称
GetDirectoryRoot	返回指定路径的卷信息、根信息或两者同时返回
GetFiles	返回指定目录中的文件的名称
GetFileSystemEntries	返回指定目录中所有文件和子目录的名称
GetLastAccessTime	返回上次访问指定文件或目录的日期和时间
GetLastWriteTime	返回上次写入指定文件或目录的日期和时间
GetParent	检索指定路径的父目录，包括绝对路径和相对路径
Move	将文件或目录及其内容移到新位置
SetCreationTime	为指定的文件或目录设置创建日期和时间
SetCurrentDirectory	将应用程序的当前工作目录设置为指定的目录
SetLastAccessTime	设置上次访问指定文件或目录的日期和时间
SetLastWriteTime	设置上次写入目录的日期和时间

DirectoryEntry 类

DirectoryEntry 类可封装 Active Directory 域服务层次结构中的节点或对象。

更多资源请访问稀酷客(www.ckook.com)

命名空间

System.DirectoryServices

属性

DirectoryEntry 类的常用属性及描述如表 D.23 所示。

表 D.23 DirectoryEntry 类的常用属性及描述

属　性	描　述
Children	获取 Active Directory 域服务层次结构中此节点的子项
Guid	获取 DirectoryEntry 的 GUID
Name	获取用基础目录服务命名的对象的名称
NativeGuid	获取提供程序返回的 DirectoryEntry 的 GUID
NativeObject	获取本机 Active Directory 服务接口（ADSI）对象
Options	获取此项的提供程序特定选项
Parent	获取 Active Directory 域服务层次结构中此项的父级
Password	设置客户端身份验证所使用的密码
Path	获取或设置此 DirectoryEntry 的路径
Properties	获取此 DirectoryEntry 对象的 Active Directory 域服务属性
SchemaClassName	获取用于此 DirectoryEntry 对象的架构类的名称
SchemaEntry	获取此项的架构对象
Username	获取或设置在对客户端进行身份验证时使用的用户名

方法

DirectoryEntry 类的常用方法及描述如表 D.24 所示。

表 D.24 DirectoryEntry 类的常用方法及描述

方　法	描　述
Close	关闭 DirectoryEntry 对象并释放与此组件关联的任何系统资源
CommitChanges	将目录项所作更改保存到基础目录存储中
CopyTo	创建此项的副本
DeleteTree	从 Active Directory 域服务层次结构中删除此项及其整个子树
Exists	确定指定路径是否表示目录服务实际项
Invoke	对本机 Active Directory 域服务对象调用方法
InvokeGet	获取来自本机 Active Directory 域服务对象的属性
InvokeSet	设置本机 Active Directory 域服务对象上的属性
MoveTo	移动此 DirectoryEntry
Rename	更改此 DirectoryEntry 对象的名称

更多资源请访问稀酷客(www.ckook.com)

DirectoryInfo 类

公开用于创建、移动和枚举目录和子目录的实例方法，无法继承此类。

说明

如果打算多次重用某个对象，可考虑使用 DirectoryInfo 的实例方法，而不是 Directory 类的相应静态方法，因为并不总是需要安全检查。

命名空间

System.IO

属性

DirectoryInfo 类的常用属性及描述如表 D.25 所示。

表 D.25　DirectoryInfo 类的常用属性及描述

属　性	描　述
CreationTime	获取或设置当前 FileSystemInfo 对象的创建时间
Exists	获取指示目录是否存在的值
Extension	获取表示文件扩展名部分的字符串
FullName	获取目录或文件的完整目录
LastAccessTime	获取或设置上次访问当前文件或目录的时间
LastWriteTime	获取或设置上次写入当前文件或目录的时间
Name	获取 DirectoryInfo 实例的名称
Parent	获取指定子目录的父目录
Root	获取路径的根部分

DirectoryName 属性

获取表示目录的完整路径的字符串。

语法

```
public string DirectoryName { get; }
```

属性值：表示目录的完整路径的字符串。

更多资源请访问稀酷客(www.ckook.com)

命名空间

System.IO

示例

本示例使用 FileInfo 对象的 DirectoryName 属性获取指定文件的完整路径。

```
FileInfo fileInfo = new FileInfo("C:\\Test.txt");
Console.WriteLine(fileInfo.DirectoryName);
```

Dispose 方法

执行与释放或重置非托管资源相关的应用程序定义的任务。

语法

```
void Dispose()
```

Dns 类

Dns 类是一个静态类，它从 Internet 域名系统（DNS）检索关于特定主机的信息。在 IPHostEntry 类的实例中返回来自 DNS 查询的主机信息。如果指定的主机在 DNS 数据库中有多个入口，则 IPHostEntry 包含多个 IP 地址和别名。

命名空间

System.Net

方法

Dns 类的常用方法及描述如表 D.26 所示。

表 D.26　Dns 类的常用方法及描述

方　法	描　述
BeginGetHostAddresses	异步返回指定主机的 Internet 协议（IP）地址
BeginGetHostByName	开始异步请求关于指定 DNS 主机名的 IPHostEntry 信息
BeginGetHostEntry	将主机名或 IP 地址异步解析为 IPHostEntry 实例
BeginResolve	开始异步请求将 DNS 主机名或 IP 地址解析为 IPAddress 实例
EndGetHostAddresses	结束对 DNS 信息的异步请求
EndGetHostByName	结束对 DNS 信息的异步请求
EndGetHostEntry	结束对 DNS 信息的异步请求

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
EndResolve	结束对 DNS 信息的异步请求
GetHostAddresses	返回指定主机的 Internet 协议（IP）地址
GetHostByAddress	获取 IP 地址的 DNS 主机信息
GetHostByName	获取指定 DNS 主机名的 DNS 信息
GetHostEntry	将主机名或 IP 地址解析为 IPHostEntry 实例
GetHostName	获取本地计算机的主机名
Resolve	将 DNS 主机名或 IP 地址解析为 IPHostEntry 实例

do…while 语句

do…while 语句与 while 语句相似，它的判断条件在循环后。do…while 语句根据条件执行一条语句一次或多次。

语法

```
do
{
    【语句块】
}while(【布尔表达式】);
```

说明

do…while 语句的执行顺序如下：

（1）程序首先执行“语句块”；

（2）当程序到达“语句块”的结束点时，计算“布尔表达式”的值，如果“布尔表达式”的值是 true，程序转到 do…while 语句的开头；否则，结束循环。

Dock 属性

获取或设置哪些控件的边框将停靠在父控件上，并确定控件如何随父控件一起调整大小。

语法

```
public virtual DockStyle Dock { get; set; }
```


更多资源请访问稀酷客(www.ckook.com)

属性值：DockStyle 枚举值之一，默认为 None。

DockStyle 枚举值的说明如表 D.27 所示。

表 D.27　DockStyle 枚举值的说明

枚 举 值	描　　述
Bottom	该控件的下边缘停靠在其包含控件的底部
Fill	控件的各个边缘分别停靠在其包含控件的各个边缘，并且适当调整大小
Left	该控件的左边缘停靠在其包含控件的左边缘
None	该控件未停靠
Right	该控件的右边缘停靠在其包含控件的右边缘
Top	该控件的上边缘停靠在其包含控件的顶端

Draw 方法

Draw 方法用来在指定的边界内和指定的表面上绘制光标。

语法

```
public void Draw (Graphics g,Rectangle targetRect)
```

Draw 方法的语法中各参数的说明如表 D.28 所示。

表 D.28　Draw 方法的语法中各参数的说明

参　　数	描　　述
g	要在其上绘制 Cursor 的 Graphics 表面
targetRect	代表 Cursor 限定范围的 Rectangle

DrawArc 方法

DrawArc 方法用来绘制圆弧。

语法

- 绘制一段弧线，它表示由 Rectangle 结构指定的椭圆的一部分

```
public void DrawArc (Pen pen,Rectangle rect,float startAngle,float
sweepAngle)
```

DrawArc 方法的语法中各参数的说明如表 D.29 所示。

- 绘制一段弧线，它表示 RectangleF 结构指定的椭圆的一部分

```
public void DrawArc(Pen pen,RectangleF rect,float startAngle,float
sweepAngle)
```


更多资源请访问稀酷客(www.ckook.com)

表 D.29　DrawArc 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定弧线的颜色、宽度和样式
rect	Rectangle 结构，它定义椭圆的边界
startAngle	从 x 轴到弧线的起始点沿顺时针方向度量的角（以度为单位）
sweepAngle	从 startAngle 参数到弧线的结束点沿顺时针方向度量的角（以度为单位）

DrawArc 方法的语法中各参数的说明如表 D.30 所示。

表 D.30　DrawArc 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定弧线的颜色、宽度和样式
rect	RectangleF 结构，它定义椭圆的边界
startAngle	从 x 轴到弧线的起始点沿顺时针方向度量的角（以度为单位）
sweepAngle	从 startAngle 参数到弧线的结束点沿顺时针方向度量的角（以度为单位）

- 绘制一段弧线，它表示由一对坐标、宽度和高度指定的椭圆部分

```
    public void DrawArc (Pen pen,int x,int y,int width,int height,int
startAngle,int sweepAngle)
    public void DrawArc(Pen pen,float x,float y,float width,float
height,float startAngle,float sweepAngle)
```

DrawArc 方法的语法中各参数的说明如表 D.31 所示。

表 D.31　DrawArc 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定弧线的颜色、宽度和样式
x	定义椭圆边框的左上角的 x 坐标
y	定义椭圆边框的左上角的 y 坐标
width	定义椭圆边框的宽度
height	定义椭圆边框的高度
startAngle	从 x 轴到弧线的起始点沿顺时针方向度量的角（以度为单位）
sweepAngle	从 startAngle 参数到弧线的结束点沿顺时针方向度量的角（以度为单位）



命名空间

System.Drawing

DrawEllipse 方法

DrawEllipse 方法用来绘制椭圆。

更多资源请访问稀酷客(www.ckook.com)

语法

- 绘制边界由 Rectangle 结构指定的椭圆

```
public void DrawEllipse (Pen pen,Rectangle rect)
```

DrawEllipse 方法的语法中各参数的说明如表 D.32 所示。

表 D.32 DrawEllipse 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定曲线的颜色、宽度和样式
rect	Rectangle 结构，它定义椭圆的边界

- 绘制边界 RectangleF 结构定义的椭圆

```
public void DrawEllipse (Pen pen,RectangleF rect)
```

DrawEllipse 方法的语法中各参数的说明如表 D.33 所示。

表 D.33 DrawEllipse 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定曲线的颜色、宽度和样式
rect	RectangleF 结构，它定义椭圆的边界

- 绘制一个由边框（该边框由一对坐标、高度和宽度指定）定义的椭圆

```
public void DrawEllipse (Pen pen,int x,int y,int width,int height)
public void DrawEllipse(Pen pen,float x,float y,float width,float
height)
```

DrawEllipse 方法的语法中各参数的说明如表 D.34 所示。

表 D.34 DrawEllipse 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定曲线的颜色、宽度和样式
x	定义椭圆边框的左上角的 x 坐标
y	定义椭圆边框的左上角的 y 坐标
width	定义椭圆边框的宽度
height	定义椭圆边框的高度

命名空间

System.Drawing

DrawImage 方法

DrawImage 方法用来绘制 Image 图像。

更多资源请访问稀酷客(www.ckook.com)

语法

- 在指定的位置按原始大小绘制指定的 Image 图像

```
public void DrawImage (Image image,Point point)
```

DrawImage 方法的语法中各参数的说明如表 D.35 所示。

表 D.35　DrawImage 方法的语法中各参数的说明

参　数	描　述
image	要绘制的 Image 图像
point	Point 结构，它表示所绘制图像的左上角的位置

- 在指定位置按指定大小绘制指定的 Image 图像

```
public void DrawImage (Image image,Rectangle rect)
```

DrawImage 方法的语法中各参数的说明如表 D.36 所示。

表 D.36　DrawImage 方法的语法中各参数的说明

参　数	描　述
image	要绘制的 Image 图像
rect	Rectangle 结构，它指定所绘制图像的位置和大小

- 在指定位置按指定大小绘制指定 Image 图像的指定部分

```
public void DrawImage (Image image,Point[] destPoints,Rectangle
srcRect,GraphicsUnit srcUnit)
```

DrawImage 方法的语法中各参数的说明如表 D.37 所示。

表 D.37　DrawImage 方法的语法中各参数的说明

参　数	描　述
image	要绘制的 Image 图像
destPoints	由 3 个 Point 结构组成的数组，这 3 个结构定义一个平行四边形
srcRect	Rectangle 结构，它指定 image 图像中要绘制的部分
srcUnit	GraphicsUnit 枚举成员，它指定 srcRect 参数所用的度量单位

命名空间

System.Drawing

DrawLine 方法

DrawLine 方法用来绘制直线。

更多资源请访问稀酷客(www.ckook.com)

语法

- 绘制一条连接两个 Point 结构的线条

```
public void DrawLine (Pen pen,Point pt1,Point pt2)
```

DrawLine 方法的语法中各参数的说明如表 D.38 所示。

表 D.38 DrawLine 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定线条的颜色、宽度和样式
pt1	Point 结构，它表示要连接的第一个点
pt2	Point 结构，它表示要连接的第二个点

- 绘制一条连接两个 PointF 结构的线条

```
public void DrawLine (Pen pen,PointF pt1,PointF pt2)
```

DrawLine 方法的语法中各参数的说明如表 D.39 所示。

表 D.39 DrawLine 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定线条的颜色、宽度和样式
pt1	PointF 结构，它表示要连接的第一个点
pt2	PointF 结构，它表示要连接的第二个点

- 绘制一条连接由坐标指定的两个点的线条

```
public void DrawLine (Pen pen,int x1,int y1,int x2,int y2)
public void DrawLine(Pen pen,float x1,float y1,float x2,float y2)
```

DrawLine 方法的语法中各参数的说明如表 D.40 所示。

表 D.40 DrawLine 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定线条的颜色、宽度和样式
x1	第一个点的 x 坐标
y1	第一个点的 y 坐标
x2	第二个点的 x 坐标
y2	第二个点的 y 坐标

命名空间

System.Drawing

DrawPath 方法

DrawPath 方法用来绘制 GraphicsPath 图形路径。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public void DrawPath (Pen pen,GraphicsPath path)
```

DrawPath 方法的语法中各参数的说明如表 D.41 所示。

表 D.41 DrawPath 方法的语法中各参数的说明

参 数	描 述
pen	Pen 对象，它确定路径的颜色、宽度和样式
path	要绘制的 GraphicsPath 图形路径

命名空间

System.Drawing

DrawPie 方法

DrawPie 方法用来绘制扇形。

语法

- 绘制由一个 Rectangle 结构和两条射线所指定的椭圆定义的扇形

```
public void DrawPie(Pen pen,Rectangle rect,float startAngle,float
sweepAngle)
```

DrawPie 方法的语法中各参数的说明如表 D.42 所示。

表 D.42 DrawPie 方法的语法中各参数的说明

参 数	描 述
pen	Pen，它确定扇形的颜色、宽度和样式
rect	Rectangle 结构，它表示定义该扇形所属的椭圆的边框
startAngle	从 x 轴到扇形的第一条边沿顺时针方向度量的角（以度为单位）
sweepAngle	从 startAngle 参数到扇形的第二条边沿顺时针方向度量的角（以度为单位）

- 绘制由一个 RectangleF 结构和两条射线所指定的椭圆定义的扇形

```
public void DrawPie(Pen pen,RectangleF rect,float startAngle,float
sweepAngle)
```

DrawPie 方法的语法中各参数的说明如表 D.43 所示。

表 D.43 DrawPie 方法的语法中各参数的说明

参 数	描 述
pen	Pen，它确定扇形的颜色、宽度和样式
rect	RectangleF 结构，它表示定义该扇形所属的椭圆的边框

更多资源请访问稀酷客(www.ckook.com)

续表

参　数	描　述
startAngle	从 x 轴到扇形的第一条边沿顺时针方向度量的角（以度为单位）
sweepAngle	从 startAngle 参数到扇形的第二条边沿顺时针方向度量的角（以度为单位）

- 绘制一个扇形，该形状由一个坐标对、宽度、高度以及两条射线所指定的椭圆定义

```
    public void DrawPie(Pen pen,int x,int y,int width,int height,int
startAngle,int sweepAngle)
    public void DrawPie (Pen pen,float x,float y,float width,float
height,float startAngle,float sweepAngle)
```

DrawPie 方法的语法中各参数的说明如表 D.44 所示。

表 D.44　DrawPie 方法的语法中各参数的说明

参　数	描　述
pen	Pen，它确定扇形的颜色、宽度和样式
x	边框的左上角的 x 坐标，该边框定义扇形所属的椭圆
y	边框的左上角的 y 坐标，该边框定义扇形所属的椭圆
width	边框的宽度，该边框定义扇形所属的椭圆
height	边框的高度，该边框定义扇形所属的椭圆
startAngle	从 x 轴到扇形的第一条边沿顺时针方向度量的角（以度为单位）
sweepAngle	从 startAngle 参数到扇形的第二条边沿顺时针方向度量的角（以度为单位）

命名空间

System.Drawing

DrawPolygon 方法

DrawPolygon 方法用来绘制多边形。

语法

- 绘制由一组 Point 结构定义的多边形

```
public void DrawPolygon (Pen pen,Point[] points)
```

DrawPolygon 方法的语法中各参数的说明如表 D.45 所示。

表 D.45　DrawPolygon 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定多边形的颜色、宽度和样式
points	Point 结构数组，这些结构表示多边形的顶点

更多资源请访问稀酷客(www.ckook.com)

- 绘制由一组 PointF 结构定义的多边形

```
public void DrawPolygon (Pen pen,PointF[] points)
```

- DrawPolygon 方法的语法中各参数的说明如表 D.46 所示。

表 D.46　DrawPolygon 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定多边形的颜色、宽度和样式
points	PointF 结构数组，这些结构表示多边形的顶点

命名空间

System.Drawing

DrawRectangle 方法

DrawRectangle 方法用来绘制矩形。

语法

- 绘制由 Rectangle 结构指定的矩形

```
public void DrawRectangle (Pen pen,Rectangle rect)
```

DrawRectangle 方法的语法中各参数的说明如表 D.47 所示。

表 D.47　DrawRectangle 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定矩形的颜色、宽度和样式
rect	表示要绘制矩形的 Rectangle 结构

- 绘制由坐标对、宽度和高度指定的矩形

```
public void DrawRectangle (Pen pen,int x,int y,int width,int height)
public void DrawRectangle(Pen pen,float x,float y,float width,float
height)
```

DrawRectangle 方法的语法中各参数的说明如表 D.48 所示。

表 D.48　DrawRectangle 方法的语法中各参数的说明

参　数	描　述
pen	Pen 对象，它确定矩形的颜色、宽度和样式
x	要绘制矩形的左上角的 x 坐标
y	要绘制矩形的左上角的 y 坐标
width	要绘制矩形的宽度
height	要绘制矩形的高度

更多资源请访问稀酷客(www.ckook.com)

命名空间

System.Drawing

DrawString 方法

DrawString 方法用来绘制文本字符串。

语法

- 在指定位置并且用指定的 Brush 和 Font 对象绘制指定的文本字符串

```
public void DrawString (string s, Font font,Brush brush,PointF point)
public void DrawString(string s,Font font,Brush brush,RectangleF layoutRectangle)
```

DrawString 方法的语法中各参数的说明如表 D.49 所示。

表 D.49 DrawString 方法的语法中各参数的说明

参 数	描 述
s	要绘制的字符串
font	Font 对象，指定字符串的文本格式
brush	Brush 对象，指定绘制文本的颜色和纹理
point	PointF 结构，指定绘制文本的左上角的坐标
layoutRectangle	RectangleF 结构，它指定所绘制文本的位置

- 使用指定 StringFormat 的格式化属性，用指定的 Brush 和 Font 对象在指定的位置绘制指定的文本字符串

```
public void DrawString(string s,Font font,Brush brush,PointF point,StringFormat format)
public void DrawString(string s,Font font,Brush brush,RectangleF layoutRectangle,StringFormat format)
```

DrawString 方法的语法中各参数的说明如表 D.50 所示。

表 D.50 DrawString 方法的语法中各参数的说明

参 数	描 述
s	要绘制的字符串
font	Font 对象，指定字符串的文本格式
brush	Brush 对象，指定绘制文本的颜色和纹理
point	PointF 结构，指定绘制文本的左上角的坐标
layoutRectangle	RectangleF 结构，它指定所绘制文本的位置
format	StringFormat，它指定应用于所绘制文本的格式化属性（如行距和对齐方式）

更多资源请访问稀酷客(www.ckook.com)

- 在指定位置并且用指定的 Brush 和 Font 对象绘制指定的文本字符串

```
public void DrawString(string s,Font font,Brush brush,float x,float y)
```

DrawString 方法的语法中各参数的说明如表 D.51 所示。

表 D.51　DrawString 方法的语法中各参数的说明

参　数	描　述
s	要绘制的字符串
font	Font 对象，指定字符串的文本格式
brush	Brush 对象，指定绘制文本的颜色和纹理
x	所绘制文本的左上角的 x 坐标
y	所绘制文本的左上角的 y 坐标

- 使用指定 StringFormat 的格式化属性，用指定的 Brush 和 Font 对象在指定的位置绘制指定的文本字符串

```
public void DrawString(string s,Font font,Brush brush,float x,float
y,StringFormat format)
```

DrawString 方法的语法中各参数的说明如表 D.52 所示。

表 D.52　DrawString 方法的语法中各参数的说明

参　数	描　述
s	要绘制的字符串
font	Font 对象，指定字符串的文本格式
brush	Brush 对象，指定绘制文本的颜色和纹理
x	所绘制文本的左上角的 x 坐标
y	所绘制文本的左上角的 y 坐标
format	StringFormat，它指定应用于所绘制文本的格式化属性（如行距和对齐方式）



命名空间

System.Drawing

DriveInfo 类

DriveInfo 类提供对有关驱动器的信息的访问。



命名空间

System.IO

更多资源请访问稀酷客(www.ckook.com)

属性

DriveInfo 类的常用属性及描述如表 D.53 所示。

表 D.53 DriveInfo 类的常用属性及描述

属 性	描 述
AvailableFreeSpace	指示驱动器上的可用空闲空间量
DriveFormat	获取文件系统的名称，例如 NTFS 或 FAT32
DriveType	获取驱动器类型
IsReady	获取一个指示驱动器是否已准备好的值
Name	获取驱动器的名称
RootDirectory	获取驱动器的根目录
TotalFreeSpace	获取驱动器上的可用空闲空间总量
TotalSize	获取驱动器上存储空间的总大小
VolumeLabel	获取或设置驱动器的卷标

方法

DriveInfo 类的常用方法及描述如表 D.54 所示。

表 D.54 DriveInfo 类的常用方法及描述

方 法	描 述
GetDrives	检索计算机上的所有逻辑驱动器的驱动器名称

DropDownStyle 属性

获取或设置指定组合框样式的值。

语法

```
public ComboBoxStyle DropDownStyle { get; set; }
```

属性值：ComboBoxStyle 枚举值之一，默认为 DropDown。

ComboBoxStyle 枚举值的说明如表 D.55 所示。

表 D.55 ComboBoxStyle 枚举值的说明

枚 举 值	描 述
DropDown	文本部分可编辑，用户必须单击箭头按钮来显示列表部分，这是默认样式
DropDownList	用户不能直接编辑文本部分，用户必须单击箭头按钮来显示列表部分
Simple	文本部分可编辑，列表部分总可见

更多资源请访问稀酷客(www.ckook.com)

迭代器

功能

迭代器是可以返回相同类型的值的有序序列的一段代码，可用做方法、运算符或 get 访问器的代码体。迭代器代码使用 yield return 语句依次返回每个元素，yield break 语句将终止迭代。可以在类中实现多个迭代器，每个迭代器都必须像任何类成员一样有唯一的名称，并且可以在 foreach 语句中被客户端代码调用。迭代器的返回类型必须为 IEnumerable 或 IEnumerator 中的任意一种。

更多资源请访问稀酷客(www.ckook.com)

E

Enabled 属性

获取或设置一个值，该值指示控件是否可以对用户交互作出响应。

语法

```
public bool Enabled { get; set; }
```

属性值：如果控件可以对用户交互作出响应，则为 true；否则为 false。默认为 true。

示例

本示例将 button1 控件设置为不可用状态。

```
button1.Enabled=false;
```

本示例将 textBox1 控件设置为不可用状态。

```
textBox1.Enabled=false;
```

本示例将 datagridView1 控件设置为不可用状态。

```
datagridView1.Enabled=false;
```

Encoding 属性

获取将输出写入到其中的 Encoding。

语法

```
public override Encoding Encoding { get; }
```

属性值：用来写入输出的 Encoding。

命名空间

System.IO

示例

本示例使用 Encoding 属性获取 StreamWriter 流写入对象的默认编码方式。

更多资源请访问稀酷客(www.ckook.com)

```
StreamWriter SWriter = new StreamWriter("C:\\Test.txt");
MessageBox.Show(SWriter.Encoding.ToString());
```

Enter 事件

鼠标进入控件时发生。

语法

```
public event EventHandler Enter
```

Environment 类

Environment 类提供有关当前环境和平台的信息以及操作它们的方法，无法继承此类。

属性

Environment 类的常用属性及描述如表 E.1 所示。

表 E.1　Environment 类的常用属性及描述

属　性	描　述
CommandLine	获取该进程的命令行
CurrentDirectory	获取或设置当前工作目录的完全限定路径
ExitCode	获取或设置进程的退出代码
HasShutdownStarted	获取一个值，该值指示是否公共语言运行库正在关闭或者当前的应用程序域正在卸载
MachineName	获取此本地计算机的 NetBIOS 名称
NewLine	获取为此环境定义的换行字符串
OSVersion	获取包含当前平台标识符和版本号的 OperatingSystem 对象
ProcessorCount	获取当前计算机上的处理器数
SystemDirectory	获取系统目录的完全限定路径
TickCount	获取系统启动后经过的毫秒数
UserDomainName	获取与当前用户关联的网络域名
UserInteractive	获取一个值，用以指示当前进程是否在用户交互模式中运行
UserName	获取当前已登录到 Windows 操作系统的人员的用户名
Version	获取一个 Version 对象，该对象描述公共语言运行库的主版本、次版本、内部版本和修订号
WorkingSet	获取映射到进程上下文的物理内存量

更多资源请访问稀酷客(www.ckook.com)

方法

Environment 类的常用方法及描述如表 E.2 所示。

表 E.2 Environment 类的常用方法及描述

方 法	描 述
Exit	终止此进程并为基础操作系统提供指定的退出代码
ExpandEnvironmentVariables	将嵌入到指定字符串中的每个环境变量的名称替换为该变量的值的等效字符串，然后返回结果字符串
FailFast	终止进程但不执行任何活动 try-finally 块或终结器
GetCommandLineArgs	返回包含当前进程的命令行参数的字符串数组
GetEnvironmentVariable	检索环境变量的值
GetEnvironmentVariables	检索所有环境变量名及其值
GetFolderPath	获取指向由指定枚举标识的系统特殊文件夹的路径
GetLogicalDrives	返回包含当前计算机中的逻辑驱动器名称的字符串数组
SetEnvironmentVariable	创建、修改或删除环境变量

Equals 方法

确定两个 String 对象是否具有相同的值。

语法

- 确定此 String 实例是否与指定的对象（也必须是 String）具有相同的值

```
public override bool Equals(Object obj)
```

Equals 方法的语法中各参数的说明如表 E.3 所示。

表 E.3 Equals 方法的语法中各参数的说明

参 数	描 述
obj	一个 Object
返回值	如果 obj 为 String 并且它的值与此实例相同，则为 true；否则为 false

- 确定此实例是否与另一个指定的 String 对象具有相同的值

```
public bool Equals(string value)
```

Equals 方法的语法中各参数的说明如表 E.4 所示。

表 E.4 Equals 方法的语法中各参数的说明

参 数	描 述
value	String
返回值	如果 value 参数的值与此实例相同，则为 true；否则为 false

更多资源请访问稀酷客(www.ckook.com)

- 确定两个指定的 String 对象是否具有同一值

```
public static bool Equals(string a,string b)
```

Equals 方法的语法中各参数的说明如表 E.5 所示。

表 E.5 Equals 方法的语法中各参数的说明

参 数	描 述
a	String 或 null
b	String 或 null
返回值	如果 a 的值与 b 的值相同，则为 true；否则为 false

- 确定此字符串是否与指定的 String 对象具有相同的值。参数指定区域性、大小写以及比较所用的排序规则

```
public bool Equals(string value,StringComparison comparisonType)
```

Equals 方法的语法中各参数的说明如表 E.6 所示。

表 E.6 Equals 方法的语法中各参数的说明

参 数	描 述
value	一个 String 对象
comparisonType	System.StringComparison 枚举值之一
返回值	如果 value 参数的值与此字符串相同，则为 true；否则为 false

StringComparison 枚举值的说明如表 E.7 所示。

表 E.7 StringComparison 枚举值的说明

枚 举 值	描 述
CurrentCulture	使用区域敏感排序规则和当前区域比较字符串
CurrentCultureIgnoreCase	使用区域敏感排序规则、当前区域来比较字符串，同时忽略被比较字符串的大小写
InvariantCulture	使用区域敏感排序规则和固定区域比较字符串
InvariantCultureIgnoreCase	使用区域敏感排序规则、固定区域来比较字符串，同时忽略被比较字符串的大小写
Ordinal	使用序号排序规则比较字符串
OrdinalIgnoreCase	使用序号排序规则并忽略被比较字符串的大小写，对字符串进行比较

- 确定两个指定的 String 对象是否具有同一值。参数指定区域性、大小写以及比较所用的排序规则

```
public static bool Equals(string a,string b,StringComparison
comparisonType)
```

Equals 方法的语法中各参数的说明如表 E.8 所示。

更多资源请访问稀酷客(www.ckook.com)

表 E.8 Equals 方法的语法中各参数的说明

参　数	描　述
a	String 对象或 null
b	String 对象或 null
comparisonType	System.StringComparison 枚举值之一
返回值	如果 a 参数的值等于 b 参数的值，则为 true；否则为 false

ErrorProvider 控件

ErrorProvider 控件可以在不打扰用户的情况下向用户显示有错误发生。当验证用户在窗体中的输入或显示数据集内的错误时，一般要用到该控件。

属性

ErrorProvider 控件的常用属性及描述如表 E.9 所示。

表 E.9 ErrorProvider 控件的常用属性及描述

属　性	描　述
BlinkRate	获取或设置错误图标的闪烁速率
BlinkStyle	获取或设置一个值，通过该值指示错误图标的闪烁时间
DataMember	获取或设置数据源中要监视的列表
DataSource	获取或设置 ErrorProvider 监视的数据源
Icon	获取或设置 Icon，当为控件设置了错误描述字符串时，该图标显示在控件旁边

方法

ErrorProvider 控件的常用方法及描述如表 E.10 所示。

表 E.10 ErrorProvider 控件的常用方法及描述

方　法	描　述
BindToDataAndErrors	提供在运行时同时设置 DataSource 和 DataMember 的方法
Clear	清除与此控件关联的所有设置
GetError	返回指定控件的当前错误描述字符串
GetIconAlignment	获取一个值，通过该值指示错误图标相对于控件的放置位置
SetError	设置指定控件的错误描述字符串
SetIconAlignment	设置错误图标相对于控件的放置位置

示例

本示例通过设置 BlinkRate 属性，指定错误图标的闪烁速率；通过设置 BlinkStyle 属性指定错误图标的闪烁时间。

更多资源请访问稀酷客(www.ckook.com)

```
this.errorProvider1.BlinkRate = 100;
this.errorProvider1.BlinkStyle = ErrorBlinkStyle.AlwaysBlink;
```

EventLog 组件

EventLog 组件可以轻松地连接到本地和远程计算机上的事件日志，并向这些日志中写入项，也可以从现有日志中读取条目并创建自己的自定义事件日志。除了向事件日志中写入项以及读取事件日志中的项之外，还可以使用“服务器资源管理器”查看有访问权限的任何服务器的事件日志列表。可以展开“服务器资源管理器”中的“事件日志”节点，以查看当前存在的日志列表，以及已写入其中的事件。此外，还可以从“服务器资源管理器”中将某些特定事件日志拖放到设计器上，以创建 EventLog 组件的实例，该实例自动将其属性设置为指向选定的日志。

命名空间

System.Diagnostics

属性

EventLog 组件的常用属性及描述如表 E.11 所示。

表 E.11 EventLog 组件的常用属性及描述

属 性	描 述
EnableRaisingEvents	获取或设置一个值，用以指示 EventLog 是否接收 EntryWritten 事件通知
Entries	获取事件日志的内容
Log	获取或设置读取或写入的日志名称
LogDisplayName	获取事件日志的好友名称
MachineName	获取或设置在其上读取或写入事件的计算机名称
MaximumKilobytes	获取或设置最大事件日志大小（以 KB 为单位）
MinimumRetentionDays	获取要在事件日志中保留项的天数
OverflowAction	获取在事件日志达到其最大日志文件大小时存储新项的已配置行为
Source	获取或设置在写入事件日志时要注册和使用的源名称

方法

EventLog 组件的常用方法及描述如表 E.12 所示。

更多资源请访问稀酷客(www.ckook.com)

表 E.12 EventLog 组件的常用方法及描述

方 法	描 述
BeginInit	开始初始化用于窗体或由其他组件使用的 EventLog。此初始化在运行时发生
Clear	从事件日志中移除所有项
Close	关闭事件日志并释放读写句柄
CreateEventSource	建立一个能够将事件信息写入到系统的特定日志中的应用程序
Delete	移除日志资源
Delete	从事件日志中移除应用程序的事件源注册
EndInit	结束用于窗体或由其他组件使用的 EventLog 的初始化。此初始化在运行时发生
Exists	确定指定的日志是否存在
GetEventLogs	创建事件日志的数组
ModifyOverflowPolicy	更改已配置的行为，以便在事件日志达到其最大文件大小时写入新项
RegisterDisplayName	指定该事件日志的本地化名称，它显示在服务器的事件查看器中
SourceExists	在计算机的注册表中搜索给定的事件源
WriteEntry	将项写入事件日志
WriteEvent	向事件日志写入本地化事件项

ExecuteNonQuery 方法

对连接执行 Transact-SQL 语句并返回受影响的行数。



语法

```
public override int ExecuteNonQuery()
```

返回值：受影响的行数。



注意

使用 ExecuteNonQuery 方法来执行目录操作（例如查询数据库的结构或创建诸如表等的数据库对象），或通过执行 UPDATE 语句、INSERT 语句或 DELETE 语句，对于 UPDATE 语句、INSERT 语句和 DELETE 语句，返回值为该命令所影响的行数。对于所有其他类型的语句，返回值为-1。如果发生回滚，返回值也为-1。

更多资源请访问稀酷客(www.ckook.com)

命名空间

System.Data.SqlClient、System.Data.Odbc、System.Data.OleDb、System.Data.OracleClient

ExecuteReader 方法

将 CommandText 属性发送到 Connection 对象并生成一个 SqlDataReader、OdbcDataReader、OleDbDataReader 或 OracleDataReader 结果集。

语法

```
public SqlDataReader ExecuteReader()
public SqlDataReader ExecuteReader(CommandBehavior behavior)
```

ExecuteReader 方法的语法中各参数的说明如表 E.13 所示。

表 E.13 ExecuteReader 方法的语法中各参数的说明

参　数	描　述
behavior	CommandBehavior 枚举值之一
返回值	一个 SqlDataReader、OdbcDataReader、OleDbDataReader 或 OracleDataReader 对象

CommandBehavior 枚举值的说明如表 E.14 所示。

表 E.14 CommandBehavior 枚举值的说明

枚 举 值	描　述
Default	此查询可能返回多个结果集，执行查询可能会影响数据库状态。Default 不设置 CommandBehavior 标志，因此调用 ExecuteReader(CommandBehavior.Default) 在功能上等效于调用 ExecuteReader()
SingleResult	查询返回一个结果集
SchemaOnly	查询仅返回列信息。当使用 SchemaOnly 时，用于 SQL Server 的.NET Framework 数据提供程序将在要执行的语句前加上 SET FMTONLY ON
KeyInfo	此查询返回列和主键信息
SingleRow	查询应返回一行，执行查询可能会影响数据库的状态。一些.NET Framework 数据提供程序可能（但不要求）使用此信息来优化命令的性能。用 OleDbCommand 对象的 ExecuteReader 方法指定 SingleRow 时，用于 OLE DB 的.NET Framework 数据提供程序使用 OLE DB IRow 接口（如果可用）执行绑定。否则，它使用 IRowset 接口。如果 SQL 语句应该只返回一行，则指定 SingleRow 还可以提高应用程序性能。在执行返回多个结果集的查询时，可以指定 SingleRow，在这种情况下，仍返回多个结果集，但每个结果集只有一行

更多资源请访问稀酷客(www.ckook.com)

续表

枚举值	描　述
SequentialAccess	提供一种方法，以便 DataReader 处理包含带有大于二进制值的列的行。SequentialAccess 不是加载整行，而是使 DataReader 将数据作为流来加载。然后可以使用 GetBytes 或 GetChars 方法来指定开始读取操作的字节位置以及正在返回的数据的有限的缓冲区大小
CloseConnection	在执行该命令时，如果关闭关联的 DataReader 对象，则关联的 Connection 对象也将关闭

命名空间

System.Data.SqlClient、System.Data.Odbc、System.Data.OleDb、System.Data.OracleClient

Exists 方法

语法

- File 类的 Exists 方法

确定指定的文件是否存在。

```
public static bool Exists(string path)
```

File 类的 Exists 方法的语法中各参数的说明如表 E.15 所示。

表 E.15　File 类的 Exists 方法的语法中各参数的说明

参　数	描　述
path	要检查的文件
返回值	如果调用方具有要求的权限并且 path 包含现有文件的名称，则为 true；否则为 false。如果 path 为 null、无效路径或零长度字符串，则此方法也将返回 false。如果调用方不具有读取指定文件所需的足够权限，则不引发异常并且该方法返回 false，这与 path 是否存在无关

- Directory 类的 Exists 方法

确定给定路径是否引用磁盘上的现有目录。

```
public static bool Exists(string path)
```

Directory 类的 Exists 方法的语法中各参数的说明如表 E.16 所示。

表 E.16　Directory 类的 Exists 方法的语法中各参数的说明

参　数	描　述
path	要测试的路径
返回值	如果 path 引用现有目录，则为 true；否则为 false

更多资源请访问稀酷客(www.ckook.com)

命名空间

System.IO

Exists 属性

语法

- FileInfo 类的 Exists 属性

获取指示文件是否存在的值。

```
public override bool Exists { get; }
```

属性值：如果该文件存在，则为 true；如果该文件不存在或如果该文件是目录，则为 false。

- DirectoryInfo 类的 Exists 属性

获取指示目录是否存在的值。

```
public override bool Exists { get; }
```

属性值：如果目录存在，则为 true；否则为 false。

- FileSystemInfo 类的 Exists 属性

获取指示文件或目录是否存在的值。

```
public abstract bool Exists { get; }
```

属性值：如果文件或目录存在，则为 true；否则为 false。

命名空间

System.IO

ExpandAll 方法

展开所有树节点。

语法

```
public void ExpandAll()
```

示例

本示例在选中 checkBox1 控件时展开 treeView1 控件中的所有树节点。

更多资源请访问稀酷客(www.ckook.com)

```
private void checkBox1_CheckedChanged(object sender, System.EventArgs e)
{
   if (checkBox1.Checked == true)
   {
      treeView1.ExpandAll();
   }
}
```

Extension 属性

获取表示文件扩展名部分的字符串。

语法

```
public string Extension { get; }
```

属性值：包含 FileSystemInfo 扩展名的字符串。

命名空间

System.IO

更多资源请访问稀酷客(www.ckook.com)

F

File 类

File 类支持对文件的基本操作，它包括用于创建、复制、删除、移动和打开文件的静态方法，并协助创建 FileStream 对象。



说明

由于 File 类中的所有方法都是静态的，所以如果只想执行一个操作，那么使用 File 类中方法的效率比使用相应的 FileInfo 类中的方法可能更高。

命名空间

System.IO

方法

File 类的常用方法及描述如表 F.1 所示。

表 F.1 File 类的常用方法及描述

方 法	描 述
Copy	将现有文件复制到新文件
Create	在指定路径中创建文件
Delete	删除指定的文件。如果指定的文件不存在，则不引发异常
Exists	确定指定的文件是否存在
Move	将指定文件移到新位置，并提供指定新文件名的选项
Open	打开指定路径上的 FileStream
CreateText	创建或打开一个文件用于写入 UTF-8 编码的文本
GetCreationTime	返回指定文件或目录的创建日期和时间
GetLastAccessTime	返回上次访问指定文件或目录的日期和时间
GetLastWriteTime	返回上次写入指定文件或目录的日期和时间
OpenRead	打开现有文件以进行读取
OpenText	打开现有 UTF-8 编码文本文件以进行读取
OpenWrite	打开现有文件以进行写入
ReadAllBytes	打开一个文件，将文件的内容读入一个字符串，然后关闭该文件

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
ReadAllLines	打开一个文本文件，将文件的所有行都读入一个字符串数组，然后关闭该文件
ReadAllText	打开一个文本文件，将文件的所有行读入一个字符串，然后关闭该文件
Replace	使用其他文件的内容替换指定文件的内容，这一过程将删除原始文件，并创建被替换文件的备份
SetCreationTime	设置创建该文件的日期和时间
SetLastAccessTime	设置上次访问指定文件的日期和时间
SetLastWriteTime	设置上次写入指定文件的日期和时间
WriteAllBytes	创建一个新文件，在其中写入指定的字节数组，然后关闭该文件。如果目标文件已存在，则改写该文件
WriteAllLines	创建一个新文件，在其中写入指定的字符串，然后关闭文件。如果目标文件已存在，则改写该文件
WwriteAllText	创建一个新文件，在文件中写入内容，然后关闭文件。如果目标文件已存在，则改写该文件

FileInfo 类

FileInfo 类和 File 类之间许多方法调用都是相同的，但是 FileInfo 类没有静态方法，该类中的方法仅可以用于实例化的对象。File 类是静态类，所以它的调用需要字符串参数为每一个方法调用规定文件位置。因此如果要在对象上进行单一方法调用，则可以使用静态 File 类，在这种情况下静态调用速度要快一些，因为.NET 框架不必执行实例化新对象并调用其方法的过程。如果要在文件上执行几种操作，则实例化 FileInfo 对象使用其方法就更好一些，这样会提高效率，因为对象将在文件系统上引用正确的文件，而静态类就必须每次都寻找文件。

命名空间

System.IO

属性

FileInfo 类的常用属性及描述如表 F.2 所示。

表 F.2　FileInfo 类的常用属性及描述

属　性	描　述
CreationTime	获取或设置当前 FileSystemInfo\对象的创建时间
Directory	获取父目录的实例

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
DirectoryName	获取表示目录的完整路径的字符串
Exists	获取指示文件是否存在的值
Extension	获取表示文件扩展名部分的字符串
FullName	获取目录或文件的完整目录
IsReadOnly	获取或设置确定当前文件是否为只读的值
LastAccessTime	获取或设置上次访问当前文件或目录的时间
LastWriteTime	获取或设置上次写入当前文件或目录的时间
Length	获取当前文件的大小
Name	获取文件名

方法

FileInfo 类的常用方法及描述如表 F.3 所示。

表 F.3　FileInfo 类的常用方法及描述

方　法	描　述
AppendText	创建一个 StreamWriter，它向 FileInfo 的此实例表示的文件追加文本
CopyTo	将现有文件复制到新文件
Create	创建文件
CreateText	创建写入新文本文件的 StreamWriter
Decrypt	使用 Encrypt 方法解密由当前账户加密的文件
Delete	永久删除文件
Encrypt	将某个文件加密，使得只有加密该文件的账户才能将其解密
MoveTo	将指定文件移到新位置，并提供指定新文件名的选项
Open	用各种读/写访问权限和共享特权打开文件
OpenRead	创建只读 FileStream
OpenText	创建使用 UTF8 编码，从现有文本文件中进行读取的 StreamReader
OpenWrite	创建只写 FileStream
Replace	使用当前 FileInfo 对象所描述的文件替换指定文件的内容，这一过程将删除原始文件，并创建被替换文件的备份

FileName 属性

获取或设置一个包含在文件对话框中选定的文件名的字符串。

语法

```
public string FileName { get; set; }
```


更多资源请访问稀酷客(www.ckook.com)

属性值：对话框中选择的文件名，默认值为空字符串（""）。

示例

本示例用来输出“打开”对话框中选择的文件名。

```
MessageBox.Show(openFileDialog1.FileName);
```

本示例使用 FileName 属性设置要打开的进程。

```
Process myProcess = new Process();
myProcess.StartInfo.FileName = "C:\\C#编程全能词典速查手册.doc";
```

FileStream 类

FileStream 类公开以文件为主的 Stream，它表示在磁盘或网络路径上指向文件的流。一个 FileStream 类的实例实际上代表一个磁盘文件，它通过 Seek 方法进行对文件的随机访问，也同时包含了流的标准输入、标准输出、标准错误等。FileStream 默认对文件的打开方式是同步的，但它同样很好地支持异步操作。

命名空间

System.IO

属性

FileStream 类的常用属性及描述如表 F.4 所示。

表 F.4 FileStream 类的常用属性及描述

属　性	描　述
CanRead	获取一个值，该值指示当前流是否支持读取
CanSeek	获取一个值，该值指示当前流是否支持查找
CanTimeout	获取一个值，该值确定当前流是否可以超时
CanWrite	获取一个值，该值指示当前流是否支持写入
IsAsync	获取一个值，该值指示 FileStream 是异步还是同步打开的
Length	获取用字节表示的流长度
Name	获取传递给构造函数的 FileStream 的名称
Position	获取或设置此流的当前位置
ReadTimeout	获取或设置一个值，该值确定流在超时前尝试读取多长时间
WriteTimeout	获取或设置一个值，该值确定流在超时前尝试写入多长时间

更多资源请访问稀酷客(www.ckook.com)

方法

FileStream 类的常用方法及描述如表 F.5 所示。

表 F.5 FileStream 类的常用方法及描述

方 法	描 述
BeginRead	开始异步读操作
BeginWrite	开始异步写操作
Close	关闭当前流并释放与之关联的所有资源
EndRead	等待挂起的异步读取完成
EndWrite	结束异步写入，在 I/O 操作完成之前一直阻止
Lock	允许读取访问的同时防止其他进程更改 FileStream
Read	从流中读取字节块并将该数据写入给定缓冲区中
ReadByte	从文件中读取一个字节，并将读取位置提升一个字节
Seek	将该流的当前位置设置为给定值
SetLength	将该流的长度设置为给定值
Unlock	允许其他进程访问以前锁定的某个文件的全部或部分
Write	使用从缓冲区读取的数据将字节块写入该流
WriteByte	将一个字节写入文件流的当前位置

Fill 方法

语法

- 在 DataSet 中添加或刷新行

```
public override int Fill(DataSet dataSet)
```

Fill 方法的语法中各参数的说明如表 F.6 所示。

表 F.6 Fill 方法的语法中各参数的说明

参 数	描 述
dataSet	要用记录和架构（如果必要）填充的 DataSet
返回值	已在 DataSet 中成功添加或刷新的行数，这不包括受不返回行的语句影响的行

- 在 DataSet 的指定范围中添加或刷新行，以与使用 DataTable 名称的数据源中的行匹配

```
public int Fill(DataTable dataTable)
```

Fill 方法的语法中各参数的说明如表 F.7 所示。

更多资源请访问稀酷客(www.ckook.com)

表 F.7 Fill 方法的语法中各参数的说明

参　数	描　述
dataTable	用于表映射的 DataTable 的名称
返回值	已在 DataSet 中成功添加或刷新的行数，这不包括受不返回行的语句影响的行

- 在 DataSet 中添加或刷新行，以匹配使用 DataSet 和 DataTable 名称的数据源中的行

```
public int Fill(DataSet dataSet,string srcTable)
```

Fill 方法的语法中各参数的说明如表 F.8 所示。

表 F.8 Fill 方法的语法中各参数的说明

参　数	描　述
dataSet	要用记录和架构（如果必要）填充的 DataSet
srcTable	用于表映射的源表的名称
返回值	已在 DataSet 中成功添加或刷新的行数，这不包括受不返回行的语句影响的行

命名空间

System.Data.SqlClient、System.Data.Odbc、System.Data.OleDb、System.Data.OracleClient

FillEllipse 方法

FillEllipse 方法用来填充边框所定义的椭圆的内部，该边框由一对坐标、一个宽度和一个高度指定。

语法

- 填充 Rectangle 结构指定的边框所定义的椭圆的内部

```
public void FillEllipse(Brush brush,Rectangle rect)
```

FillEllipse 方法的语法中各参数的说明如表 F.9 所示。

表 F.9 FillEllipse 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
rect	Rectangle 结构，它表示定义椭圆的边框

- 填充 RectangleF 结构指定的边框所定义的椭圆的内部

更多资源请访问稀酷客(www.ckook.com)

```
public void FillEllipse(Brush brush,RectangleF rect)
```

FillEllipse 方法的语法中各参数的说明如表 F.10 所示。

表 F.10　FillEllipse 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
rect	RectangleF 结构，它表示定义椭圆的边框

- 填充边框所定义的椭圆的内部，该边框由一对坐标、一个宽度和一个高度指定

```
public void FillEllipse(Brush brush,int x,int y,int width,int
height)
public void FillEllipse(Brush brush,float x,float y,float
width,float height)
```

FillEllipse 方法的语法中各参数的说明如表 F.11 所示。

表 F.11　FillEllipse 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
x	定义椭圆的边框的左上角的 X 坐标
y	定义椭圆的边框的左上角的 Y 坐标
width	定义椭圆的边框的宽度
height	定义椭圆的边框的高度

命名空间

System.Drawing

FillPath 方法

FillPath 方法用来填充 GraphicsPath 的内部。

语法

```
public void FillPath(Brush brush,GraphicsPath path)
```

FillPath 方法的语法中各参数的说明如表 F.12 所示。

表 F.12　FillPath 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
path	GraphicsPath，它表示要填充的路径

更多资源请访问稀酷客(www.ckook.com)

命名空间

System.Drawing

FillPie 方法

FillPie 方法用来填充扇形。

语法

- 填充椭圆所定义的扇形区的内部，该椭圆由 Rectangle 结构和两条射线指定

```
public void FillPie(Brush brush,Rectangle rect,float
startAngle,float sweepAngle)
```

FillPie 方法的语法中各参数的说明如表 F.13 所示。

表 F.13 FillPie 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
rect	Rectangle 结构，它表示定义该扇形区所属的椭圆的边框
startAngle	从 x 轴沿顺时针方向旋转到扇形区第一个边所测得的角度（以度为单位）
sweepAngle	从 startAngle 参数沿顺时针方向旋转到扇形区第二个边所测得的角度（以度为单位）

- 填充由一对坐标、一个宽度、一个高度以及两条射线指定的椭圆所定义的扇形区的内部

```
public void FillPie(Brush brush,int x,int y,int width,int height,int
startAngle,int sweepAngle)
public void FillPie(Brush brush,float x,float y,float width,float
height,float startAngle,float sweepAngle)
```

FillPie 方法的语法中各参数的说明如表 F.14 所示。

表 F.14 FillPie 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
x	边框左上角的 x 坐标，该边框定义扇形区所属的椭圆
y	边框左上角的 y 坐标，该边框定义扇形区所属的椭圆
width	边框的宽度，该边框定义扇形区所属的椭圆
height	边框的高度，该边框定义扇形区所属的椭圆
startAngle	从 x 轴沿顺时针方向旋转到扇形区第一个边所测得的角度（以度为单位）
sweepAngle	从 startAngle 参数沿顺时针方向旋转到扇形区第二个边所测得的角度（以度为单位）

更多资源请访问稀酷客(www.ckook.com)

命名空间

System.Drawing

FillPolygon 方法

FillPolygon 方法用来填充 Point 或 PointF 结构指定的点数组所定义的多边形的内部。

语法

- 填充 Point 结构指定的点数组所定义的多边形的内部

```
public void FillPolygon(Brush brush,Point[] points)
```

FillPolygon 方法的语法中各参数的说明如表 F.15 所示。

表 F.15 FillPolygon 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
points	Point 结构数组，这些结构表示要填充的多边形的顶点

- 填充 PointF 结构指定的点数组所定义的多边形的内部

```
public void FillPolygon(Brush brush,PointF[] points)
```

FillPolygon 方法的语法中各参数的说明如表 F.16 所示。

表 F.16 FillPolygon 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
points	PointF 结构数组，这些结构表示要填充的多边形的顶点

- 使用指定的填充模式填充 Point 结构指定的点数组所定义的多边形的内部

```
public void FillPolygon(Brush brush,Point[] points,FillMode
fillMode)
```

FillPolygon 方法的语法中各参数的说明如表 F.17 所示。

表 F.17 FillPolygon 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
points	Point 结构数组，这些结构表示要填充的多边形的顶点
fillMode	确定填充样式的 FillMode 枚举的成员

更多资源请访问稀酷客(www.ckook.com)

FillMode 枚举值的说明如表 F.18 所示。

表 F.18　FillMode 枚举值的说明

枚举值	描　述
Alternate	指定交替填充模式
Winding	指定环绕填充模式

- 使用指定的填充模式填充 PointF 结构指定的点数组所定义的多边形的内部

```
public void FillPolygon(Brush brush,PointF[] points,FillMode
fillMode)
```

FillPolygon 方法的语法中各参数的说明如表 F.19 所示。

表 F.19　FillPolygon 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
points	PointF 结构数组，这些结构表示要填充的多边形的顶点
fillMode	确定填充样式的 FillMode 枚举的成员

命名空间

System.Drawing

FillRectangle 方法

FillRectangle 方法用来填充一个矩形框的内部。

语法

- 填充 Rectangle 结构指定的矩形的内部

```
public void FillRectangle(Brush brush,Rectangle rect)
```

FillRectangle 方法的语法中各参数的说明如表 F.20 所示。

表 F.20　FillRectangle 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
rect	Rectangle 结构，它表示要填充的矩形

- 填充 RectangleF 结构指定的矩形的内部

```
public void FillRectangle(Brush brush,RectangleF rect)
```

FillRectangle 方法的语法中各参数的说明如表 F.21 所示。

更多资源请访问稀酷客(www.ckook.com)

表 F.21　FillRectangle 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
rect	RectangleF 结构，它表示要填充的矩形

- 填充由一对坐标、一个宽度和一个高度指定的矩形的内部

```
    public void FillRectangle(Brush brush,int x,int y,int width,int
height)
    public void FillRectangle(Brush brush,float x,float y,float
width,float height)
```

FillRectangle 方法的语法中各参数的说明如表 F.22 所示。

表 F.22　FillRectangle 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
x	要填充的矩形的左上角的 x 坐标
y	要填充的矩形的左上角的 y 坐标
width	要填充的矩形的宽度
height	要填充的矩形的高度

命名空间

System.Drawing

FillRegion 方法

FillRegion 方法用来填充 Region 的内部。

语法

```
    public void FillRegion(Brush brush,Region region)
```

FillRegion 方法的语法中各参数的说明如表 F.23 所示。

表 F.23　FillRegion 方法的语法中各参数的说明

参　数	描　述
brush	确定填充特性的 Brush
region	Region，它表示要填充的区域

命名空间

System.Drawing

更多资源请访问稀酷客(www.ckook.com)

Filter 属性

获取或设置当前文件名筛选器字符串，该字符串决定对话框的“另存为文件类型”或“文件类型”框中出现的选择内容。

语法

```
public string Filter { get; set; }
```

属性值：对话框中可用的文件筛选选项。

示例

本示例用来设置“打开”对话框中只能选择各种格式的图片文件。

```
openFileDialog1.Filter = "*.jpg,*.jpeg,*.bmp,*.gif,*.ico,*.png,
*.tif,*.wmf|*.jpg;*.jpeg;*.bmp;*.gif;*.ico;*.png;*.tif;*.wmf";
```

本示例用来设置在“保存”对话框中只能保存 txt 文本文件。

```
saveFileDialog1.Filter = "文本文件(*.txt)|*.txt";
saveFileDialog1.ShowDialog();
```

Find 方法

语法

- 在 RichTextBox 控件中搜索指定文本

```
public int Find (string str)
```

RichTextBox.Find 方法的语法中各参数的说明如表 F.24 所示。

表 F.24　RichTextBox.Find 方法的语法中各参数的说明

参　数	描　述
str	要在控件中定位的文本
返回值	在控件中找到搜索文本的位置；如果未找到搜索字符串或者在 str 参数中指定了空搜索字符串，则为-1

- 搜索名称与指定键匹配的项，也可以选择搜索子项

```
public ListViewItem[] Find(string key,bool searchAllSubItems)
```

ListView 控件的 Find 方法的语法中各参数的说明如表 F.25 所示。

更多资源请访问稀酷客(www.ckook.com)

表 F.25　ListView 控件的 Find 方法的语法中各参数的说明

参　数	描　述
key	要搜索的项名称
searchAllSubItems	若要搜索子项，则为 true；否则为 false
返回值	ListViewItem 类型数组

- 搜索与指定谓词所定义的条件相匹配的元素，并返回整个 List<T>中的第一个匹配元素

```
public T Find(Predicate<T> match)
```

List<T>.Find 方法的语法中各参数的说明如表 F.26 所示。

表 F.26　List<T>.Find 方法的语法中各参数的说明

参　数	描　述
match	Predicate <T>委托，用于定义要搜索的元素的条件
返回值	如果找到与指定谓词定义的条件匹配的第一个元素，则为该元素；否则为类型 T 的默认值



说明

List<T>.Find 方法在 System.Collections.Generic 命名空间下。

First 函数

First 函数用于返回在查询所返回的结果集中的第一个记录的字段值。



语法

```
First(expression)
```

First 函数的语法中各参数的说明如表 F.27 所示。

表 F.27　First 函数的语法中各参数的说明

参　数	描　述
expression	代表一个字符串表达式，它标识了包含要使用数据的字段，或者是使用该字段中数据执行计算的表达式。expression 中的操作数可以包括表字段、常量或函数（可以是固有的或者用户自定义的函数，但不能是其他 SQL 聚合函数）的名称

更多资源请访问稀酷客(www.ckook.com)

FirstDayOfWeek 属性

根据月历中的显示获取或设置一周中的第一天。

语法

```
public Day FirstDayOfWeek { get; set; }
```

属性值：Day 枚举值之一。默认为 Day. Default。

Day 枚举值的说明如表 F.28 所示。

表 F.28 Day 枚举值的说明

枚举值	描述
Monday	星期一
Tuesday	星期二
Wednesday	星期三
Thursday	星期四
Friday	星期五
Saturday	星期六
Sunday	星期日
Default	由应用程序指定的一周中默认的某天

示例

本示例中将月历控件中的每周的第一天设置为星期一。

```
this.monthCalendar1.FirstDayOfWeek = Day.Monday;
```

FlatStyle 属性

获取或设置控件的平面样式外观。

语法

```
public FlatStyle FlatStyle { get; set; }
```

属性值：FlatStyle 枚举值之一，默认值为 Standard。

FlatStyle 枚举值的说明如表 F.29 所示。

表 F.29 FlatStyle 枚举值的说明

枚举值	描述
Flat	该控件以平面显示

更多资源请访问稀酷客(www.ckook.com)

续表

枚举值	描　述
Popup	该控件以平面显示，直到鼠标指针移动到该控件为止，此时该控件外观为三维
Standard	该控件外观为三维
System	该控件的外观是由用户的操作系统决定的

FlowDirection 属性

获取或设置一个值，该值指示 FlowLayoutPanel 控件的流向。



语法

```
public FlowDirection FlowDirection { get; set; }
```

属性值：FlowDirection 枚举值之一，指示控件在面板中的连续放置方向，默认为 LeftToRight。

FlowDirection 枚举值的说明如表 F.30 所示。

表 F.30　FlowDirection 枚举值的说明

枚举值	描　述
LeftToRight	元素从设计图面的左边缘流到右边缘
TopDown	元素从设计图面的顶部流到底部
RightToLeft	元素从设计图面的右边缘流到左边缘
BottomUp	元素从设计图面的底部流到顶部



示例

本示例设置 FlowLayoutPanel 控件中的元素布局方向为从底部到顶部。

```
flowLayoutPanel1.FlowDirection = FlowDirection.BottomUp;
```

FlowLayoutPanel 控件

FlowLayoutPanel 控件用来在水平或垂直方向排列其内容。开发人员可以将 FlowLayoutPanel 控件的内容从一行换至下一行，或者从一列换至下一列，还可以选择剪裁内容而不是换行。任何 Windows 窗体控件都可以是 FlowLayoutPanel 控件的子控件，包括 FlowLayoutPanel 的其他实例，利用此功能，可以在运行时构造适应窗体尺寸大小的复杂布局。

更多资源请访问稀酷客(www.ckook.com)

属性

FlowLayoutPanel 控件的常用属性及描述如表 F.31 所示。

表 F.31　FlowLayoutPanel 控件的常用属性及描述

属　性	描　述
FlowDirection	指示控件布局的方向
WrapContents	指定是换行还是剪裁 FlowLayoutPanel 控件的内容
AutoSize	使 FlowLayoutPanel 控件自动调整大小以容纳其内容
FlowBreak	设置为 true 会使 FlowLayoutPanel 控件停止在当前流方向布局控件并换到下一行或下一列

方法

FlowLayoutPanel 控件的常用方法及描述如表 F.32 所示。

表 F.32　FlowLayoutPanel 控件的常用方法及描述

方　法	描　述
GetFlowBreak	返回一个值，该值表示控件的流中断设置
SetFlowBreak	设置一个值，该值表示控件的流中断设置

Flush 方法

语法

- Stream 类的 Flush 方法

当在派生类中重写时，将清除该流的所有缓冲区，并使得所有缓冲数据被写入到基础设备。

```
public abstract void Flush()
```

- FileStream 类的 Flush 方法

清除该流的所有缓冲区会使得所有缓冲的数据都将写入到文件系统。

```
public override void Flush()
```

- CryptoStream 类的 Flush 方法

清除该流的所有缓冲区，使得所有缓冲的数据都被写入到基础设备。

```
public override void Flush()
```

- RegistryKey 类的 Flush 方法

将指定的打开注册表项的全部属性写到注册表中。

```
public void Flush()
```


更多资源请访问稀酷客(www.ckook.com)

命名空间

System.IO、System.Security.Cryptography、Microsoft.Win32

Focus 方法

为控件设置输入焦点。

语法

```
public bool Focus()
```

返回值：如果输入焦点请求成功，则为 true；否则为 false。

示例

本示例在文本框中输入数据后，按回车键即可使下一个文本框获得输入焦点。

```
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 13)          //判断是否按下回车键
    {
        textBox2.Focus( );        //把焦点移到文本框 2
    }
}
```

FolderBrowserDialog 控件

FolderBrowserDialog 控件提示用户浏览、创建并最终选择一个文件夹。如果只允许用户选择文件夹而非文件，则可使用此控件。文件夹的浏览通过树控件完成。

说明

使用 FolderBrowserDialog 控件时，只能选择文件系统中的文件夹；不能选择虚拟文件夹。

属性

FolderBrowserDialog 控件的常用属性及描述如表 F.33 所示。

更多资源请访问稀酷客(www.ckook.com)

表 F.33 FolderBrowserDialog 控件的常用属性及描述

属　性	描　述
Description	获取或设置对话框中在树视图控件上显示的说明文本
RootFolder	获取或设置从其开始浏览的根文件夹
SelectedPath	获取或设置用户选定的路径
ShowNewFolderButton	获取或设置一个值，该值指示“新建文件夹”按钮是否显示在文件夹浏览对话框中

方法

FolderBrowserDialog 控件的常用方法及描述如表 F.34 所示。

表 F.34 FolderBrowserDialog 控件的常用方法及描述

方　法	描　述
RunDialog	在派生类中被重写时，指定通用对话框
ShowDialog	运行通用对话框

Font 属性

语法

- Control 类的 Font 属性

获取或设置控件显示的文字的字体。

```
public virtual Font Font { get; set; }
```

属性值：要应用于由控件显示的文本的 Font。默认为 DefaultFont 属性的值。

- FontDialog 控件的 Font 属性

获取或设置选定的字体。

```
public Font Font { get; set; }
```

属性值：选定的字体。

示例

本示例将 textBox1 控件中显示的字体设置为 fontDialog1 控件中选中的字体。

```
textBox1.Font = fontDialog1.Font;
```

本示例设置字体对话框中的默认字体为指定的字体。

```
fontDialog1.Font = new Font("宋体",12,FontStyle.Bold);
```


更多资源请访问稀酷客(www.ckook.com)

FontDialog 控件

FontDialog控件用于公开系统上当前安装的字体，开发人员可在 Windows 应用程序中将其用做简单的字体选择解决方案，而不是配置自己的对话框。默认情况下，“字体”对话框显示字体、字体样式和字体大小的列表框、删除线和下画线等效果的复选框、脚本（脚本是指给定字体可用的不同字符脚本，如希伯来语或日语等）的下拉列表以及字体外观等选项。

属性

FontDialog 控件的常用属性及描述如表 F.35 所示。

表 F.35　FontDialog 控件的常用属性及描述

属　性	描　述
AllowSimulations	获取或设置一个值，该值指示对话框是否允许图形设备接口（GDI）字体模拟
AllowVectorFonts	获取或设置一个值，该值指示对话框是否允许选择矢量字体
AllowVerticalFonts	获取或设置一个值，该值指示对话框是既显示垂直字体又显示水平字体，还是只显示水平字体
Color	获取或设置选定字体的颜色
FixedPitchOnly	获取或设置一个值，该值指示对话框是否只允许选择固定间距字体
Font	获取或设置选定的字体
FontMustExist	获取或设置一个值，该值指示对话框是否指定当用户试图选择不存在的字体或样式时的错误条件
MaxSize	获取或设置用户可选择的最大磅值
MinSize	获取或设置用户可选择的最小磅值
Options	获取用来初始化 FontDialog 的值
ScriptsOnly	获取或设置一个值，该值指示对话框是否允许为所有非 OEM 和 Symbol 字符集以及 ANSI 字符集选择字体
ShowApply	获取或设置一个值，该值指示对话框是否包含“应用”按钮
ShowColor	获取或设置一个值，该值指示对话框是否显示颜色选择
ShowEffects	获取或设置一个值，该值指示对话框是否包含允许用户指定删除线、下画线和文本颜色选项的控件
ShowHelp	获取或设置一个值，该值指示对话框是否显示“帮助”按钮

方法

FontDialog 控件的常用方法及描述如表 F.36 所示。

更多资源请访问稀酷客(www.ckook.com)

表 F.36　FontDialog 控件的常用方法及描述

方　法	描　述
RunDialog	指定文件对话框
ShowDialog	运行通用对话框

for 语句

for 语句用于计算一个初始化序列，然后当某个条件为真时，重复执行嵌套语句并计算一个迭代表达式序列；如果为假，则终止 for 循环。



语法

```
for(【初始化表达式】;【条件表达式】;【迭代表达式】)
{
    【语句块】
}
```



说明

“初始化表达式”由一个局部变量声明或者由一个逗号分隔的表达式列表组成，用“初始化表达式”声明的局部变量的作用域从变量的声明开始，一直到嵌入语句的结尾；“条件表达式”必须是一个布尔表达式；“迭代表达式”必须包含一个用逗号分隔的表达式列表。

foreach 语句

foreach 语句用于枚举一个集合的元素，并对该集合中的每个元素执行一次嵌入语句，但是 foreach 语句不应用于更改集合内容，以避免产生不可预知的错误。



语法

```
foreach(【类型】 【迭代变量名】 in 【集合类型表达式】)
{
    【语句块】
}
```


更多资源请访问稀酷客(www.ckook.com)



说明

“类型”和“迭代变量名”用于声明迭代变量，迭代变量相当于一个范围覆盖整个语句块的局部变量。在 foreach 语句执行期间，迭代变量表示当前正在为其执行迭代的集合元素。“集合类型表达式”必须有一个从该集合的元素类型到迭代变量的类型的显示转换，如果“集合类型表达式”的值为 null，则会出现异常。

ForeColor 属性

获取或设置控件的前景色。



语法

```
public virtual Color ForeColor { get; set; }
```

属性值：控件的前景色 Color，默认为 DefaultForeColor 属性的值。



示例

本示例将 textBox1 控件的前景色修改为系统颜色中的 Window 色彩。

```
textBox1.ForeColor = System.Drawing.SystemColors.Window;
```

本示例将 comboBox1 控件的前景色修改为黑色。

```
comboBox1.ForeColor = Color.Black;
```

Form 窗体

在 Form 窗体中，“窗体”是向用户显示信息的可视化界面。通常情况下，通过向窗体上添加控件（如单击按钮或者向文本框中录入数据），生成 Form 窗体应用程序。Form 窗体包含可添加到窗体上的各种控件：文本框、按钮、下拉框和单选按钮等。

Form 窗体具有丰富的 UI 控件，可模拟像 Microsoft Office 那样的高端应用程序中的功能。



属性

Form 窗体的常用属性及描述如表 F.37 所示。

更多资源请访问稀酷客(www.ckook.com)

表 F.37 Form 窗体的常用属性及描述

属　性	描　述
AcceptButton	获取或设置当用户按回车键时所单击的窗体上的按钮
ActiveControl	获取或设置窗体上的活动控件
ActiveForm	获取此应用程序的当前活动窗体
ActiveMdiChild	获取当前活动的多文档界面（MDI）子窗口
AllowTransparency	获取或设置一个值，该值指示能否调整窗体的不透明度
CancelButton	获取或设置当用户按 Esc 键时单击的按钮控件
ControlBox	获取或设置一个值，该值指示在该窗体的标题栏中是否显示控件框
DesktopBounds	获取或设置 Windows 桌面上窗体的大小和位置
DesktopLocation	获取或设置 Windows 桌面上窗体的位置
DialogResult	获取或设置窗体的对话框结果
Enabled	获取或设置一个值，该值指示窗体是否可以对用户交互作出响应
FormBorderStyle	获取或设置窗体的边框样式
HelpButton	获取或设置一个值，该值指示是否应在窗体的标题框中显示“帮助”按钮
Icon	获取或设置窗体的图标
IsMdiChild	获取一个值，该值指示窗体是否为多文档界面（MDI）子窗体
IsMdiContainer	获取或设置一个值，该值指示窗体是否为多文档界面（MDI）子窗体的容器
MainMenuStrip	获取或设置窗体的主菜单容器
MaximizeBox	获取或设置一个值，该值指示是否在窗体的标题栏中显示“最大化”按钮
MaximizedBounds	获取和设置窗体最大化后的大小
MaximumSize	获取窗体可调整到的最大大小
MdiChildren	获取窗体的数组，这些窗体表示以此窗体作为父级的多文档界面（MDI）子窗体
MdiParent	获取或设置此窗体的当前多文档界面（MDI）父窗体
Menu	获取或设置在窗体中显示的 MainMenu
MergedMenu	获取窗体的合并菜单
MinimizeBox	获取或设置一个值，该值指示是否在窗体的标题栏中显示“最小化”按钮
MinimumSize	获取或设置窗体可调整到的最小大小
Name	获取或设置窗体的名称
Opacity	获取或设置窗体的不透明度级别
OwnedForms	获取 Form 对象的数组，这些对象表示此窗体拥有的所有窗体
Owner	获取或设置拥有此窗体的窗体
ShowIcon	获取或设置一个值，该值指示是否在窗体的标题栏中显示图标
ShowInTaskbar	获取或设置一个值，该值指示是否在 Windows 任务栏中显示窗体
StartPosition	获取或设置运行时窗体的起始位置
Text	获取或设置窗体的标题
TopMost	获取或设置一个值，指示该窗体是否应显示为最顶层窗体

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
Visible	获取或设置一个值，该值指示是否显示该控件及其所有父控件
WindowState	获取或设置窗体的窗口状态

方法

Form 窗体的常用方法及描述如表 F.38 所示。

表 F.38　Form 窗体的常用方法及描述

方　法	描　述
Activate	激活窗体并给予它焦点
ActivateMdiChild	激活窗体的 MDI 子级
AddOwnedForm	向此窗体添加附属窗体
CenterToParent	使窗体在父窗体边界内居中
CenterToScreen	使窗体在当前屏幕上居中
Close	关闭窗体
Dispose	释放由窗体使用的资源
Hide	隐藏窗体
LayoutMdi	在 MDI 父窗体内排列多文档界面（MDI）子窗体
PointToClient	将指定屏幕点的位置计算成工作区坐标
PointToScreen	将指定工作区点的位置计算成屏幕坐标
Show	显示窗体
ShowDialog	将窗体显示为模式对话框

Format 方法

语法

- 将指定的 String 中的格式项替换为指定的 Object 实例的值的文本等效项

```
public static string Format(string format,Object arg0)
```

Format 方法的语法中各参数的说明如表 F.39 所示。

表 F.39　Format 方法的语法中各参数的说明

参　数	描　述
format	复合格式字符串
arg0	要格式化的 Object
返回值	format 的一个副本，其中的第一个格式项已替换为 arg0 的 String 等效项

更多资源请访问稀酷客(www.ckook.com)

- 将指定 String 中的格式项替换为指定数组中相应 Object 实例的值的文本等效项

```
public static string Format(string format,Object[] args)
```

Format 方法的语法中各参数的说明如表 F.40 所示。

表 F.40　Format 方法的语法中各参数的说明

参　数	描　述
format	复合格式字符串
args	包含零个或多个要格式化的对象的 Object 数组
返回值	format 的一个副本，其中格式项已替换为 args 中相应 Object 实例的 String 等效项

- 将指定 String 中的格式项替换为指定数组中相应 Object 实例的值的文本等效项。指定的参数提供区域性特定的格式设置信息

```
public static string Format(IFormatProvider provider,string
format,Object[] args)
```

Format 方法的语法中各参数的说明如表 F.41 所示。

表 F.41　Format 方法的语法中各参数的说明

参　数	描　述
provider	一个 IFormatProvider，它提供区域性特定的格式设置信息
format	复合格式字符串
args	包含零个或多个要格式化的对象的 Object 数组
返回值	format 的一个副本，其中格式项已替换为 args 中相应 Object 实例的 String 等效项

- 将指定的 String 中的格式项替换为两个指定的 Object 实例的值的文本等效项

```
public static string Format(string format,Object arg0,Object arg1)
```

Format 方法的语法中各参数的说明如表 F.42 所示。

表 F.42　Format 方法的语法中各参数的说明

参　数	描　述
format	复合格式字符串
arg0	第一个要格式化的 Object
arg1	第二个要格式化的 Object
返回值	format 的一个副本，其中的第一个和第二个格式项已替换为 arg0 和 arg1 的 String 等效项

- 将指定的 String 中的格式项替换为 3 个指定的 Object 实例的值的文本等效项

更多资源请访问稀酷客(www.ckook.com)

```
public static string Format(string format,Object arg0,Object
arg1,Object arg2)
```

Format 方法的语法中各参数的说明如表 F.43 所示。

表 F.43 Format 方法的语法中各参数的说明

参　数	描　述
format	复合格式字符串
arg0	第一个要格式化的 Object
arg1	第二个要格式化的 Object
arg2	第三个要格式化的 Object
返回值	format 的一个副本，其中的第一个、第二个和第三个格式项已替换为 arg0、arg1 和 arg2 的 String 等效项

FormClosed 事件

关闭窗体后发生。

语法

```
public event FormClosedEventHandler FormClosed
```

示例

本示例在关闭窗体后退出应用程序。

```
private void Form1_FormClosed(object sender, FormClosingEventArgs e)
{
    Application.Exit();
}
```

FormClosing 事件

关闭窗体前发生。

语法

```
public event FormClosingEventHandler FormClosing
```

FromArgb 方法

语法

- 从一个 32 位 ARGB 值创建 Color 结构

更多资源请访问稀酷客(www.ckook.com)

```
public static Color FromArgb(int argb)
```

FromArgb 方法的语法中各参数的说明如表 F.44 所示。

表 F.44 FromArgb 方法的语法中各参数的说明

参　数	描　述
argb	指定 32 位 ARGB 值的值
返回值	此方法创建的 Color 结构

- 从指定的 Color 结构创建 Color 结构，但要使用新指定的 alpha 值。尽管此方法允许为 alpha 值传递 32 位值，但该值仅限于 8 位

```
public static Color FromArgb(int alpha,Color baseColor)
```

FromArgb 方法的语法中各参数的说明如表 F.45 所示。

表 F.45 FromArgb 方法的语法中各参数的说明

参　数	描　述
alpha	新 Color 的 alpha 值，有效值为从 0 到 255
baseColor	从中创建新 Color 的 Color
返回值	此方法创建的 Color

- 从指定的 8 位颜色值（红色、绿色和蓝色）创建 Color 结构。alpha 值默认为 255（完全不透明）。尽管此方法允许为每个颜色分量传递 32 位值，但每个分量的值仅限于 8 位

```
public static Color FromArgb(int red,int green,int blue)
```

FromArgb 方法的语法中各参数的说明如表 F.46 所示。

表 F.46 FromArgb 方法的语法中各参数的说明

参　数	描　述
red	新 Color 的红色分量值，有效值为从 0 到 255
green	新 Color 的绿色分量值，有效值为从 0 到 255
blue	新 Color 的蓝色分量值，有效值为从 0 到 255
返回值	此方法创建的 Color

- 从 4 个 ARGB 分量（alpha、红色、绿色和蓝色）值创建 Color 结构。尽管此方法允许为每个分量传递 32 位值，但每个分量的值仅限于 8 位

```
public static Color FromArgb(int alpha,int red,int green,int blue)
```

FromArgb 方法的语法中各参数的说明如表 F.47 所示。

表 F.47 FromArgb 方法的语法中各参数的说明

参　数	描　述
alpha	alpha 分量，有效值为从 0 到 255
red	红色分量，有效值为从 0 到 255

更多资源请访问稀酷客(www.ckook.com)

续表

参　数	描　述
Green	绿色分量，有效值为从 0 到 255
blue	蓝色分量，有效值为从 0 到 255
返回值	此方法创建的 Color

命名空间

System.Drawing

泛型

功能

泛型是用于处理算法、数据结构的一种编程方法。泛型的目标是采用广泛适用和可交互性的形式来表示算法和数据结构，以使它们能够直接用于软件构造。泛型类、结构、接口、委托和方法可以根据它们存储和操作的数据的类型来进行参数化。泛型能在编译时，提供强大的类型检查，减少数据类型之间的显示转换、装箱操作和运行时的类型检查。泛型类和泛型方法同时具备可重用性、类型安全和效率，这是非泛型类和非泛型方法无法具备的。泛型通常用在集合和在集合上运行的方法中。

示例

本示例通过定义一个泛型方法，查找数组中某个数字的位置，示例运行结果为：“6 在数组中的位置：5”。

```
public class Finder                          //建立一个公共类 Finder
{
    public static int Find<T>(T[] items, T item) //创建泛型方法
    {
        for (int i = 0; i < items.Length; i++)    //调用 for 循环
        {
            if (items[i].Equals(item))    //调用 Equals 方法比较两个数
            {
                return i;                 //返回相等数在数组中的位置
            }
        }
        return -1;                        //如果不存在指定的数，则返回-1
    }
}
```


更多资源请访问稀酷客(www.ckook.com)

```
class Program
{
    static void Main(string[] args)
    {
        //调用泛型方法，并定义数组指定数字
        int i = Finder.Find<int>(new int[] {1, 2, 3, 4, 5, 6, 8, 9 }, 6);
        //输出中数字在数组中的位置
        Console.WriteLine("6 在数组中的位置：" + i.ToString());
        Console.ReadLine();
    }
}
```

分部类

功能

分部类的出现使得程序的结构更加合理，代码的组织更加紧密。开发人员可以将类、结构或接口的定义拆分到两个或多个源文件中，每个源文件包含类定义的一部分，编译应用程序时 Visual Studio 开发环境会把所有部分组合起来，这样的类被称为分部类。分部类是 C#2.0 中的新概念，其主要应用在以下两个方面：

- 当项目比较庞大时，使用分部类可以拆分一个类至几个文件中，这样的处理可以使得不同的开发人员同时进行工作，避免了效率的低下。
- 使用自动生成的源时，无需重新创建源文件便可将代码添加到类中。Visual Studio 在创建 Windows 窗体和 Web 服务包装代码等时都使用此方法，开发人员无需编辑 Visual Studio 所创建的文件，即可创建使用这些类的代码。

定义分部类时需要使用 partial 关键字，分部类的每个部分都必须包含一个 partial 关键字，并且其声明必须与其他部分位于同一命名空间，即分部类的各个部分只能在同一程序集和同一模块（.exe 或.dll 文件）中进行定义，而不能跨越多个模块。

典型应用

本实例通过使用分部类创建 4 个方法，它们分别用于执行加、减、乘、除运算，并返回运算后的结果。实例运行结果如图 F.1 所示。

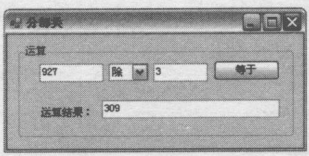


图 F.1　分部类应用实例

更多资源请访问稀酷客(www.ckook.com)

实现关键代码如下：

```
partial class account                              //分部类第一部分
{
    public int addition(int a, int b)              //创建一个整型方法
    {
        return a + b;                              //方法中的加法运算
    }
}
partial class account                              //分部类第二部分
{
    public int multiplication(int a, int b)        //创建一个整型方法
    {
        return a * b;                              //方法中的乘法运算
    }
}
partial class account                              //分部类第三部分
{
    public int subtration(int a,int b)             //创建一个整型方法
    {
        return a-b;                                //方法中的减法运算
    }
}
partial class account                              //分部类第四部分
{
    public int division(int a, int b)              //创建一个整型方法
    {
        return a / b;                              //方法中的除法运算
    }
}
private void Form1_Load(object sender, EventArgs e)
{
comboBox1.SelectedIndex = 0;
//设置 comboBox1 控件的 DropDownStyle 属性使其显示为下拉列表的样式
    comboBox1.DropDownStyle = ComboBoxStyle.DropDownList;
}
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        account at = new account();                //实例化分部类
        //获取第一个文本框中的值
        int M = int.Parse(txtNo1.Text.Trim());
//获取第二个文本框中的值
```


更多资源请访问稀酷客(www.ckook.com)

```
            int N = int.Parse(txtNo2.Text.Trim());
            //获取 comboBox1 控件选择的值
            string str = comboBox1.Text;
            switch (str)
            {
               //调用分部类中的加法运算
    case "加": txtResult.Text = at.addition(M, N).ToString(); break;
    //调用分部类中的减法运算
    case "减": txtResult.Text = at.subtration(M, N).ToString(); break;
    //调用分部类中的乘法运算
    case "乘": txtResult.Text = at.multiplication(M, N).ToString();
break;
    //调用分部类中的除法运算
              case "除": txtResult.Text = at.division(M, N).ToString();
break;
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
```


更多资源请访问稀酷客(www.ckook.com)

GDI+

GDI+是GDI（即Windows早期版本中附带的Graphics Device Interface）的后继者，它是一种构成Windows XP操作系统的子系统的应用程序编程接口（API）。一般来说，有3种基本类型的绘图界面，分别为：Windows窗体上的控件、要发送给打印机的页面和内存中的位图、图像，而Graphics类封装了一个GDI+绘图界面，因此该类提供了可以在以上3种绘图界面上绘图的功能，另外，程序开发人员还可以使用该类绘制文本、线条、矩形、曲线、多边形、椭圆、圆弧和贝塞尔样条等。

概述

GDI+指的是.NET Framework 2.0中提供的二维图形、图像处理等功能。GDI+在GDI（较早版本的Windows中提供的Graphics Device Interface）的基础上进行了改进，添加了新功能并改进了现有功能。GDI+主要用于在窗体上绘制各种图形图像，可以用于绘制各种数据图形、数学仿真等。GDI+可以在窗体程序中产生很多自定义的图形，便于开发人员展示各种图形化的数据。

GetBytes 方法

GetBytes方法在派生类中重写时，将指定的String中的所有字符编码为一个字节序列。它返回一个字节数组，包含对指定的字符集进行编码的结果。

语法

```
public virtual byte[] GetBytes (string s)
```

GetBytes方法的语法中各参数的说明如表G.1所示。

表G.1　GetBytes方法的语法中各参数的说明

参　数	描　述
s	包含要编码的字符的String
返回值	一个字节数组，包含对指定的字符集进行编码的结果

GetDirectories 方法

GetDirectories方法获取指定目录中子目录的名称。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public static string[] GetDirectories (string path)
```

GetDirectories 方法的语法中各参数的说明如表 G.2 所示。

表 G.2 GetDirectories 方法的语法中各参数的说明

参　数	描　述
path	为其返回子目录名称的数组的路径
返回值	一个类型 String 的数组，它包含 path 中子目录的名称

GetFiles 方法

Directory 类的 GetFiles()方法返回指定目录中的文件的名称。

语法

```
public static string[] GetFiles (string path)
```

GetFiles 方法的语法中各参数的说明如表 G.3 所示。

表 G.3 GetFiles 方法的语法中各参数的说明

参　数	描　述
path	指定文件的路径
返回值	指定目录中文件名的 String 数组，文件名包含完整路径

GetFileSystemEntries 方法

GetFileSystemEntries 方法返回指定目录中所有文件和子目录的名称。

语法

```
public static string[] GetFileSystemEntries (string path)
```

GetFileSystemEntries 方法的语法中各参数的说明如表 G.4 所示。

表 G.4 GetFileSystemEntries 方法的语法中各参数的说明

参　数	描　述
path	为其返回文件名和子目录名的目录
返回值	一个 String 数组，它包含指定目录中文件系统项的名称

GetFileSystemInfos 方法

GetFileSystemInfos 方法返回 DirectoryInfo 类型的实例包含的文件或目录。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public FileSystemInfo[] GetFileSystemInfos ()
```

返回值：FileSystemInfo[]，强类型 FileSystemInfo 项的数组。

GetHostAddresses 方法

DNS 类的 GetHostAddresses 方法返回指定主机的 Internet 协议（IP）地址。

语法

```
public static IPAddress[] GetHostAddresses (string
hostNameOrAddress)
```

GetHostAddresses 方法的语法中各参数的说明如表 G.5 所示。

表 G.5 GetHostAddresses 方法的语法中各参数的说明

参　数	描　述
hostNameOrAddress	要解析的主机名或 IP 地址
返回值	一个 IPAddress 类型的数组，该类型保存由 hostNameOrAddress 参数指定的主机的 IP 地址

GetHostByAddress 方法

获取 IP 地址的 DNS 主机信息，重载方法为 Dns.GetHostByAddress (IPAddress) 和 Dns.GetHostByAddress (String)。

语法

（1）Dns.GetHostByAddress (IPAddress)

根据指定的 IPAddress 创建 IPHostEntry 实例。

语法：

```
public static IPHostEntry GetHostByAddress(IPAddress address)
```

GetHostByAddress(IPAddress)方法的语法中各参数的说明如表 G.6 所示。

表 G.6 GetHostByAddress(IPAddress)方法的语法中各参数的说明

参　数	描　述
address	一个 IPAddress
返回值	一个 IPHostEntry

更多资源请访问稀酷客(www.ckook.com)

（2）Dns.GetHostByAddress (String)

根据 IP 地址创建 IPHostEntry 实例。

语法：

```
public static IPHostEntry GetHostByAddress(string address)
```

GetHostByAddress(String)方法的语法中各参数的说明如表 G.7 所示。

表 G.7　GetHostByAddress(String)方法的语法中各参数的说明

参　数	描　述
address	IP 地址
返回值	一个 IPHostEntry

GetHostByName 方法

DNS 类的 GetHostName 方法用于获取本地计算机的主机名。

语法

```
public static string GetHostName ()
```

返回值：包含本地计算机的 DNS 主机名的字符串。

GetHostName 方法

DNS 类的 GetHostName 方法用于获取本地计算机的主机名。

语法

```
public static string GetHostName ()
```

返回值：包含本地计算机的 DNS 主机名的字符串。

示例

本示例使用 DNS 类的 GetHostName 方法获取本地主机名。

```
string hostName = Dns.GetHostName();
Console.WriteLine("Computer name :" + hostName);
```

GetPixel 方法

该方法主要用来获取 Bitmap 图像中指定像素的颜色。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public Color GetPixel (int x,int y)
```

GetPixel 方法的语法中各参数的说明如表 G.8 所示。

表 G.8 GetPixel 方法的语法中各参数的说明

参　数	描　述
x	要检索的像素的 *x* 坐标
y	要检索的像素的 *y* 坐标
返回值	Color 结构，它表示指定像素的颜色

GetString 方法

此方法将指定字节数组中的所有字节解码为一个字符串。

语法

```
public virtual string GetString (byte[] bytes)
```

GetString 方法的语法中各参数的说明如表 G.9 所示。

表 G.9 GetString 方法的语法中各参数的说明

参　数	描　述
bytes	包含要解码的字节序列的字节数组
返回值	包含指定字节序列解码结果的 String

GetValueNames 方法

RegistryKey 类中主要封装了对视窗系统注册表的基本操作，例如检索指定的子项、获取包含当前项所对应值名称的字符串数组等。此类中的 GetValueNames 方法用于检索包含与此项关联的所有值名称的字符串数组。

语法

```
public string[] GetValueNames ()
```

返回值：包含当前项的值名称的字符串数组。

goto 语句

goto 语句属于无条件的跳转语句，因为 C#允许代码行加上标签，这样就

更多资源请访问稀酷客(www.ckook.com)

可以用 goto 语句直接跳转到这些代码行上。该语句有其优缺点。优点在于，可以方便的控制什么时候去执行哪些代码，主要缺点在于，过多使用这个技巧将很难读懂代码段。

语法

```
goto 标识符;
goto case 常量表达式;
goto default;
```

标识符：标签的名称。

常量表达式：一个 switch/case 语句的标签。

Graphics 类

绘制图形要求有一个绘制表面、用于定位和对齐形状的坐标系统，以及完成绘图的工具。GDI+将这些功能封装在 System.Drawing.Graphics 类中。可以使用该类的静态方法为图像和控件创建 Graphics 对象，并且，可以用其实例方法绘制各种形状，如圆、三角形和文本等。如果用户对 Win32 图形 API 很熟悉的话，就会发现它与 GDI 的设备上下文（device context）非常相似，但 Graphics 对象是一个更简单的设计。设备上下文是一个维护绘图状态信息的结构，可以作为参数传递给绘图函数。Graphics 表示绘图表面，并提供了在绘图表面上进行绘图的方法。

属性

Graphics 类的公共属性及说明如表 G.10 所示。

表 G.10　Graphics 类的公共属性及说明

属　性	说　明
Clip	获取或设置 Region 对象，该对象限定此 Graphics 对象的绘图区域
ClipBounds	获取 RectangleF 结构，该结构限定此 Graphics 对象的剪辑区域
CompositionMode	获取一个值，该值指定如何将合成图像绘制到此 Graphics 对象
CompositionQuality	获取或设置绘制到此 Graphics 对象的合成图像的呈现质量
DpiX	获取此 Graphics 对象的水平分辨率
DpiY	获取此 Graphics 对象的垂直分辨率
InterpolationMode	获取或设置与此 Graphics 对象关联的插补模式
IsClipEmpty	获取一个值，该值指示此 Graphics 对象的剪辑区域是否为空
IsVisibleClipEmpty	获取一个值，该值指示此 Graphics 对象的可见剪辑区域是否为空
PageScale	获取或设置此 Graphics 对象的全局单位和页单位之间的比例

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	说　明
PageUnit	获取或设置用于此 Graphics 对象中的页坐标的度量单位
PixelOffsetMode	获取或设置一个值，该值指定在呈现此 Graphics 对象的过程中像素如何偏移
RenderingOrigin	为抵色处理和阴影画笔获取或设置此 Graphics 对象的呈现原点
SmoothingMode	获取或设置此 Graphics 对象的呈现质量
TextContrast	获取或设置呈现文本的伽马纠正值
TextRenderingHint	获取或设置与此 Graphics 对象关联的文本的呈现模式
Transform	获取或设置此 Graphics 对象的全局变换
VisibleClipBounds	获取或设置此 Graphics 对象的可见剪辑区域的边框



方法

Graphics 类的公共方法及说明如表 G.11 所示。

表 G.11　Graphics 类的公共方法及说明

方　法	说　明
Clear	清除整个绘图面并以指定背景色填充
Dispose	释放由此 Graphics 对象使用的所有资源
DrawArc	绘制一段弧线，它表示由一对坐标、宽度和高度指定的椭圆部分
DrawBezier	绘制由 4 个 Point 结构定义的贝塞尔样条
DrawBeziers	从 Point 结构的数组绘制一系列贝塞尔样条
DrawClosedCurve	绘制由 Point 结构的数组定义的闭合基数样条
DrawCurve	绘制经过一组指定的 Point 结构的基数样条
DrawEllipse	绘制一个由边框（该边框由一对坐标、高度和宽度指定）定义的椭圆
DrawIcon	在指定坐标处绘制由指定的 Icon 对象表示的图像
DrawIconUnstretched	绘制指定的 Icon 对象表示的图像，而不缩放该图像
DrawImage	在指定位置并且按原始大小绘制指定的 Image 对象
DrawImageUnscaled	在坐标对所指定的位置并且按其原始大小绘制指定的 Image 对象
DrawLine	绘制一条连接由坐标对象指定的两个点的线条
DrawLines	绘制一列连接一组 Point 结构的线段
DrawPath	绘制 GraphicsPath 对象
DrawPie	绘制一个扇形，该扇形由一个坐标对象、宽度和高度以及两条射线所指定的椭圆定
DrawPolygon	绘制由一组 Point 结构定义的多边形
DrawRectangle	绘制由坐标对象、宽度和高度指定的矩形
DrawRectangles	绘制一系列由 Rectangle 结构指定的矩形
DrawString	在指定位置并且用指定的 Brush 和 Font 对象绘制指定的文本字符串
FillEllipse	填充边框所定义的椭圆的内部，该边框由一对坐标、一个宽度和一个高度指定

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	说　明
FillPath	填充 GraphicsPath 对象的内部
FillPie	填充由一对坐标、一个宽度、一个高度以及两条射线指定的椭圆所定义的扇形区的内部
FillPolygon	填充 Point 结构指定的点数组所定义的多边形的内部
FillRectangle	填充由一对坐标、一个宽度和一个高度指定的矩形的内部
FillRectangles	填充由 Rectangle 结构指定的一系列矩形的内部
FillRegion	填充 Region 对象的内部
Flush	强制执行所有挂起的图形操作并立即返回而不等待操作完成
FromHdc	从设备上下文的指定句柄创建新的 Graphics 对象
FromHdcInternal	内部方法。不要使用
FromHwnd	从窗口的指定句柄创建新的 Graphics 对象
FromHwndInternal	内部方法。不要使用
FromImage	从指定的 Image 对象创建新 Graphics 对象
GetHalftonePalette	获取当前 Windows 的半色调调色板的句柄
GetHdc	获取与此 Graphics 对象关联的设备上下文的句柄
IsVisible	指示由一对坐标指定的点是否包含在此 Graphics 对象的可见剪辑区域内
MeasureCharacterRanges	获取 Region 对象的数，其中每个对象将字符位置的范围限定在指定字符串内
MeasureString	测量用指定的 Font 对象绘制的指定字符串
MultiplyTransform	将此 Graphics 对象的全局变换乘以指定的 Matrix 对象
ResetTransform	将此 Graphics 对象的全局变换矩阵重置为单位矩阵
Save	保存此 Graphics 对象的当前状态，并用 GraphicsState 对象标识保存的状态

典型应用

本例通过 Graphics 类对象绘制饼形图分析公司男女比例，饼图是一种常用的用来显示数据比例的图表技术，它在突出某一重要数据时非常有用。本节将通过一个实例来介绍，如何使用饼图显示公司员工的男女比例情况。如图 G.1 所示。

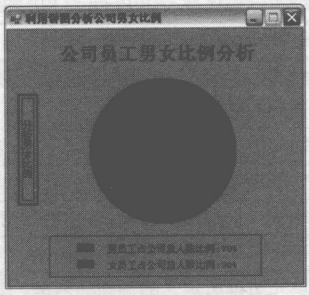


图 G.1　饼图分析男女比例

更多资源请访问稀酷客(www.ckook.com)

主要代码如下：

```
    private void button1_Click(object sender, EventArgs e)
    {
        //连接数据库
        SqlConnection sqlcon = new SqlConnection("Data Source=(local);
Database=db_14;Uid=sa;Pwd=");
        string P_str_sel = "select *  from tb_02";
        SqlDataAdapter myda1 = new SqlDataAdapter(P_str_sel, sqlcon);
        DataSet myds1 = new DataSet();
        myda1.Fill(myds1);
        //计算男、女总和
        string P_str_sumNum = "select sum(boyNum+girlNum) as sumNum FROM
tb_02";
       SqlDataAdapter myda2 = new SqlDataAdapter(P_str_sumNum, sqlcon);
       DataSet myds2 = new DataSet();
       myda2.Fill(myds2);
       int P_int_sum = Convert.ToInt32(myds2.Tables[0].Rows[0][0].ToString());
       //获取男员工数目
       int P_int_boy = Convert.ToInt32(myds1.Tables[0].Rows[0][0].ToString());
        //获取女员工数目
        int P_int_girl = Convert.ToInt32(myds1.Tables[0].Rows[0][1].
ToString());
        Graphics graphics = this.CreateGraphics();
        try
        {
            graphics.Clear(Color.White);
            Pen pen1 = new Pen(Color.Red);
            Brush brush1 = new SolidBrush(Color.YellowGreen);
            Brush brush2 = new SolidBrush(Color.Blue);
            Brush brush3 = new SolidBrush(Color.Brown);
            Font font1 = new Font("Courier New", 16, FontStyle.Bold);
            Font font2 = new Font("Courier New", 8);
            graphics.FillRectangle(brush1, 0, 0, 370, 350);
graphics.DrawString("公司员工男女比例分析", font1, brush2, new Point(60,
20));
            //男员工在圆中分配的角度
            float angle1 = Convert.ToSingle((360 * (Convert.ToSingle
(P_int_boy) / Convert.ToSingle(P_int_sum))));
            //女员工在圆中分配的角度
            float angle2 = Convert.ToSingle((360 * (Convert.ToSingle
(P_int_girl) / Convert.ToSingle(P_int_sum))));
           //绘制男员工所占比例
            graphics.FillPie(brush2, 100, 60, 180, 180, 0, angle1);
```


更多资源请访问稀酷客(www.ckook.com)

```
        //绘制女员工所占比例
         graphics.FillPie(brush3, 100, 60, 180, 180, angle1, angle2);
         graphics.DrawRectangle(pen1, 50, 255, 140, 50);  //绘制范围框
         graphics.FillRectangle(brush2, 85, 145, 20, 10);  //绘制小矩形
         graphics.DrawString("男员工占公司总人数比例:" + Convert.ToInt32
(P_int_boy) * 100 / Convert.ToInt32(P_int_sum) + "%", font2, brush2, 120,
145);
        graphics.FillRectangle(brush3, 85, 285, 20, 10);
        graphics.DrawString("女员工占公司总人数比例:" + Convert.ToInt32
(P_int_girl) * 100 / Convert.ToInt32(P_int_sum) + "%", font2, brush3, 120,
285);
      }
      catch (Exception ex)
      {
         MessageBox.Show(ex.Message, "信息", MessageBoxButtons.OK,
MessageBoxIcon.Information);
       }
       graphics.Dispose();
    }
```

GroupBox 控件

GroupBox 控件用于为其他控件提供可识别的分组。通常，使用 GroupBox 控件细分窗体。当移动单个 GroupBox 控件时，它包含的所有控件也会一起移动。



属性

GroupBox 控件常用属性及说明如表 G.12 所示。

表 G.12　GroupBox 控件常用属性及说明

属　性	说　明
TabIndex	控件的容器的控件的 Tab 键顺序
FlatStyle	该属性获取或设置组框控件的平面样式外观
Text	该属性用来设置控件的文本



方法

GroupBox 控件常用方法及说明如表 G.13 所示。

更多资源请访问稀酷客(www.ckook.com)

表 G.13　GroupBox 控件常用方法及说明

方　法	说　明
Show	向用户显示控件
SuspendLayout	临时挂起控件的布局逻辑
Select	激活控件
Hide	对用户隐藏控件
FindForm	检索控件所在的窗体

GetValue 方法

Registry 类的 GetValue 方法用于检索与指定的注册表项中的指定名称关联的值。如果在指定的项中未找到该名称，则返回您提供的默认值；或者，如果指定的项不存在，则返回空引用。



语法

```
public static Object GetValue (string keyName,string
valueName,Object defaultValue)
```

GetValue 方法的语法中各参数的说明如表 G.14 所示。

表 G.14　GetValue 方法的语法中各参数的说明

参　数	描　述
keyName	以有效注册表根（如“HKEY_CURRENT_USER”）开头的键的完整注册表路径
ValueName	名称/值对的名称
DefaultValue	当 name 不存在时返回的值
返回值	如果由 keyName 指定的子项不存在，则返回空引用；否则，返回与 valueName 关联的值；或者，如果未找到 valueName，则返回 defaultValue

更多资源请访问稀酷客(www.ckook.com)

H

Handle 属性

Control 对象的 Handle 属性用于获取控件绑定到的窗口句柄。



语法

```
public IntPtr Handle { get; }
```

属性值：包含控件的窗口句柄。

Hashtable 类

Hashtable（哈希表）是一种键/值对集合，这些键/值对根据键的哈希代码进行组织。在一个 Hashtable 中插入一对 Key/Value 时，它自动将 Key 值映射到 Value，并允许获取与一个指定的 Key 相关联的 value。在 Hashtable 中主要使用 Add、Remove 两个方法对哈希表进行操作。Add 方法用于将带有指定键和值的元素添加到 Hashtable 中；Remove 方法用于从 Hashtable 中移除带有指定键的元素。



语法

```
    //使用默认的初始容量、加载因子、哈希代码提供程序和比较器来初始化 Hashtable 类的实例
    public Hashtable();
    //使用指定容量、默认加载因子、默认哈希代码提供程序和比较器来初始化 Hashtable 类的实例
    public Hashtable(int capacity);
    //使用指定的容量，加载因子来初始化 Hashtable 类的实例
    public Hashtable(int capacity,float loadFactor);
    //通过将指定字典中的元素复制到新的 Hashtable 对象中，初始化 Hashtable 类的一个新实例。新 Hashtable 对象的初始容量等于复制的元素数，并且使用默认的加载因子、哈希代码提供程序和比较器
    public Hashtable(IDictionary d);
```

构造哈希表时，有一些新的概念需要先理解。

- 在哈希表中，键被转换为哈希代码，而值存储在存储器中。
- 初始容量指哈希表中元素的数目。加载因子是哈希表元素与存储器的最大比率，当初始容量需要自动扩展前，确定值与存储器之间的最大比率。值越小，出现冲突也越小。.NET 默认加载因子为 1.0。

更多资源请访问稀酷客(www.ckook.com)

属性

Hashtable 类的常用的属性及说明如表 H.1 所示。

表 H.1　Hashtable 类的常用的属性及说明

属　性	说　明
Count	获取包含在 Hashtable 中的键/值对的数目
IsFixedSize	获取一个值，该值指示 Hashtable 是否具有固定大小
IsReadOnly	获取一个值，该值指示 Hashtable 是否为只读
IsSynchronized	获取一个值，该值指示是否同步对 Hashtable 的访问
Item	获取或设置与指定的键相关联的值
Keys	获取包含 Hashtable 中的键的 ICollection
SyncRoot	获取可用于同步 Hashtable 访问的对象
Values	获取包含 Hashtable 中的值的 ICollection

（1）Count 属性

获取包含在 Hashtable 中的键/值对的数目。

语法：

```
public virtual int Count { get; }
```

属性值：包含在 Hashtable 中的键/值对的数目。

（2）Item 属性

获取或设置与指定的键相关联的值。

语法：

```
public virtual Object this [Object key] { get; set; }
```

参数说明：

key：要获取或设置其值的键。

属性值：与指定的键相关联的值。如果未找到指定的键，尝试获取它将返回空引用，尝试设置它将使用指定的键创建新元素。

（3）Keys 属性

获取包含 Hashtable 中的键的 ICollection。

语法：

```
public virtual ICollection Keys { get; }
```

属性值：包含 Hashtable 中的键的 ICollection。

（4）Values 属性

获取包含 Hashtable 中的值的 ICollection。

更多资源请访问稀酷客(www.ckook.com)

语法：

```
public virtual ICollection Values { get; }
```

属性值：一个 ICollection，它包含 Hashtable 中的值。

方法

Hashtable 类的常用的方法及说明如表 H.2 所示。

表 H.2 Hashtable 类的常用的方法及说明

方　法	说　明
Add	将带有指定键和值的元素添加到 Hashtable 中
Clear	从 Hashtable 中移除所有元素
Clone	创建 Hashtable 的浅表副本
Contains	确定 Hashtable 是否包含特定键
ContainsKey	确定 Hashtable 是否包含特定键
ContainsValue	确定 Hashtable 是否包含特定值
CopyTo	将 Hashtable 元素复制到一维 Array 实例中的指定索引位置
GetEnumerator	返回循环访问 Hashtable 的 IDictionaryEnumerator
GetHashCode	用做特定类型的哈希函数 GetHashCode
Remove	从 Hashtable 中移除带有指定键的元素
Synchronized	返回 Hashtable 的同步包装

（1）Add 方法

将带有指定键和值的元素添加到 Hashtable 中。

语法：

```
public virtual void Add (Object key,Object value)
```

Add 方法的语法中各参数的说明如表 H.3 所示。

表 H.3 Add 方法的语法中各参数的说明

参　数	描　述
key	要添加的元素的键
value	要添加的元素的值，该值可以为空引用

（2）Clear 方法

从 Hashtable 中移除所有元素。

语法：

```
public virtual void Clear ()
```

例如，实例化一个 Hashtable 对象，然后向其中添加数据，最后通过 Clear 方法移除所有的元素，代码如下：

更多资源请访问稀酷客(www.ckook.com)

```
Hashtable myHT = new Hashtable();
myHT.Add( "one", "The" );
myHT.Add( "two", "quick" );
myHT.Add( "three", "brown" );
myHT.Add( "four", "fox" );
myHT.Add( "five", "jumped" );
Console.WriteLine( "Initially," );
Console.WriteLine( "   Count    : {0}", myHT.Count );
Console.WriteLine( "   Values:" );
PrintKeysAndValues( myHT );
myHT.Clear();
```

（3）ContainsKey 方法

确定 Hashtable 是否包含特定键。

语法：

```
public virtual bool ContainsKey (Object key)
```

ContainsKey 方法的语法中各参数的说明如表 H.4 所示。

表 H.4　ContainsKey 方法的语法中各参数的说明

参　数	描　述
key	要在 Hashtable 中定位的键
返回值	如果 Hashtable 包含具有指定键的元素，则为 True；否则为 False

HasRows 属性

HasRows 属性用来获取一个值，用于指示 OleDbDataReader 是否包含一行或多行。

语法

```
public override bool HasRows { get; }
```

属性值：如果 SqlDataReader 包含一行或多行，则为 true；否则为 false。

Height 属性

Control 对象的 Height 属性用于获取或设置控件的高度。

语法

```
public int Height { get; set; }
```


更多资源请访问稀酷客(www.ckook.com)

属性值：控件的高度（以像素为单位）。

HelpProvider 组件

Windows 窗体 HelpProvider 组件用于将 HTML 帮助文件（.htm 文件或由 HTML Help Workshop 产生的.chm 文件）与 Windows 应用程序相关联。可以通过多种方式提供帮助：

- 为 Windows 窗体中的控件提供区分上下文的帮助。
- 为特定对话框或对话框中的特定控件提供区分上下文的帮助。
- 打开帮助文件到特定部分，如目录、索引或搜索功能的主页。

在 Windows 窗体中添加 HelpProvider 组件可以使该窗体中的其他控件公开 HelpProvider 组件的 Help 属性。这样就可以提供有关 Windows 窗体控件的帮助。可以使用 HelpNamespace 属性将帮助文件与 HelpProvider 组件相关联。通过调用 SetHelpNavigator 并为指定控件提供 HelpNavigator 枚举值来指定所提供的帮助类型。通过调用 SetHelpKeyword 方法为帮助提供关键字或主题。也可以使用 SetHelpString 方法将特定的帮助字符串与另一个控件相关联。当控件有焦点且用户按 F1 键时，在弹出式窗口中显示使用上述方法与该控件关联的字符串。如果尚未设置 HelpNamespace，则必须使用 SetHelpString 提供帮助文本。如果同时设置了 HelpNamespace 和帮助字符串，则基于 HelpNamespace 的帮助优先。



注意

如果在 HelpProvider 控件的 ShowHelp 方法或 HelpNamespace 属性中指定了帮助文件的路径，则使用相对路径时可能会遇到问题。因此，一定要使用绝对文件路径来指定帮助文件。



属性

HelpProvider 组件常用的属性及说明如表 H.5 所示。

表 H.5　HelpProvider 组件常用的属性及说明

属　性	说　明
HelpNamespace	获取或设置一个值，该值指定与此 HelpProvider 对象关联的帮助文件名
Tag	获取或设置包含有关 HelpProvider 的补充数据的对象

（1）Tag 属性

该属性获取或设置包含有关 HelpProvider 的补充数据的对象。

更多资源请访问稀酷客(www.ckook.com)

（2）HelpNamespace 属性

该属性用于获取或设置一个值，该值指定与此 HelpProvider 对象关联的帮助文件名。

语法：

```
public virtual string HelpNamespace { get; set; }]
```

属性值：帮助文件的名称。其形式可以是“C:\path\sample.chm”或“/folder/file.htm”。

例如，设置 HelpProvider 组件的 HelpNamespace 属性，代码如下：

```
string strPath = Application.StartupPath.Substring(0,
Application.StartupPath.Substring(0,
Application.StartupPath.LastIndexOf("\\")).LastIndexOf("\\"));
strPath += @"\sl\08\30\F1.html";
this.helpProvider1.HelpNamespace = strPath;
```



方法

HelpProvider 组件常用的方法及说明如表 H.6 所示。

表 H.6　HelpProvider 组件常用的方法及说明

方　法	说　明
CanExtend	指定此对象是否可以将其扩展程序属性提供给指定的对象
GetHelpKeyword	返回指定控件的帮助关键字
GetHelpString	返回指定控件的弹出帮助窗口的内容
SetHelpKeyword	指定在用户调用指定控件的帮助信息时用于检索帮助的关键字
SetHelpString	指定与指定的控件关联的帮助字符串
SetShowHelp	指定是否显示指定控件的帮助信息

下面对比较重要的方法进行详细介绍。

（1）GetHelpKeyword 方法

返回指定控件的帮助关键字。

语法：

```
public virtual string GetHelpKeyword (Control ctl)
```

GetHelpKeyword 方法的语法中各参数的说明如表 H.7 所示。

表 H.7　GetHelpKeyword 方法的语法中各参数的说明

参　数	描　述
ctl	从中检索帮助主题的 Control。返回值：与此控件关联的帮助关键字，如果 HelpProvider 组件当前配置为显示整个帮助文件或配置为提供帮助字符串，则为空引用

更多资源请访问稀酷客(www.ckook.com)

（2）SetShowHelp 方法

指定是否显示指定控件的帮助信息。

语法：

```
public virtual void SetShowHelp (Controlctl,bool value)
```

SetShowHelp 方法的语法中各参数的说明如表 H.8 所示。

表 H.8　SetShowHelp 方法的语法中各参数的说明

参　数	描　述
ctl	Control 其帮助信息已打开或关闭
Value	如果显示控件的帮助信息，则为 true；否则为 false

Hide 方法

Hide 方法可以被应用到控件可窗体上，当应用到控件上时，等效于将控件的 Visible 属性设置为 false。在调用 Hide 方法之后，只要不调用 Show 方法，Visible 属性就返回 false 值。



语法

```
public void Hide ()
```

HScrollBar 控件

HScrollBar 控件表示一个标准 Windows 水平滚动条。多数需要滚动条的控件本身提供滚动条，而不需要该控件。例如，多行的 TextBox 控件、ListBox 和 ComboBox 都是如此。可以使用此控件在本身不提供滚动条的容器（如 PictureBox）中实现滚动，或为用户输入数值数据实现滚动。数值数据可以在控件中显示，也可以在代码中使用。Minimum 和 Maximum 属性确定用户可以选择的值范围。LargeChange 属性决定在滚动条之内但在滚动框之外单击的效果。SmallChange 属性确定单击控件各端的滚动箭头的效果。



属性

HScrollBar 组件常用的属性及说明如表 H.9 所示。

表 H.9　HScrollBar 组件常用的属性及说明

属　性	说　明
AutoScrollOffset	获取或设置一个值，该值指示在 ScrollControlIntoView 中将控件滚动到何处
Height	获取或设置控件的高度

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	说　明
LargeChange	获取或设置一个值，当滚动框长距离移动时向 Value 属性加上该值或从中减去该值
Location	获取或设置该控件的左上角相对于其容器的左上角的坐标
Margin	获取或设置控件之间的空间
Maximum	获取或设置可滚动范围的上限值
MaximumSize	获取或设置大小，该大小是 GetPreferredSize 可以指定的上限
Minimum	获取或设置可滚动范围的值的下限
MinimumSize	获取或设置大小，该大小是 GetPreferredSize 可以指定的下限
Name	获取或设置控件的名称
Size	获取或设置控件的高度和宽度
SmallChange	获取或设置小距离移动滚动框时，在 Value 属性中加上或减去的值
Value	获取或设置表示滚动框在滚动条控件中的当前位置的数值
Visible	获取或设置一个值，该值指示是否显示该控件

（1）Value 属性

Value 属性（默认情况下为 0）是与滚动框在滚动条中的位置相对应的一个 integer 值。当滚动框的位置值为最小值时，滚动框移到最左端位置（对于水平滚动条），或移到顶端位置（对于垂直滚动条）。当滚动框的位置值为最大值时，滚动框移到最右端位置或底端位置。同理，在值范围的底端和顶端中间的值会使滚动框的前端位于滚动条的中间。

除了可以通过鼠标单击来更改滚动条值以外，用户还可以沿着滚动条将滚动框拖动到任何点。结果值取决于滚动框的位置，但是该值总是在用户设置的 Minimum 属性和 Maximum 属性的范围之内。

语法：

```
[BindableAttribute(true)]
public int Value { get; set; }
```

属性值：处于 Minimum 和 Maximum 范围内的数值。默认值为 0。

（2）LargeChange 属性

当用户按下 Page Up 键或 Page Down 键或者在滚动框的任何一边单击滚动条轨迹时，Value 属性将按照 LargeChange 属性中设置的值而更改。

语法：

```
public int LargeChange { get; set; }
```

属性值：一个数值。默认值为 10。

更多资源请访问稀酷客(www.ckook.com)

（3）SmallChange 属性

当用户按下某个箭头键或单击某个滚动条按钮时，Value 属性将按照 SmallChange 属性中设置的值而更改。

语法：

```
public int SmallChange { get; set; }
```

属性值：一个数值。默认值为 1。

更多资源请访问稀酷客(www.ckook.com)

I

IsUpper 方法

判断字符串中某一字符是否大写时，用到了 Char 类的 IsUpper 方法，该方法用来判断指定字符串中位于指定位置处的字符是否为大写字母。

语法

```
public static bool IsUpper (string s,int index)
```

IsUpper 方法的语法中各参数的说明如表 I.1 所示。

表 I.1 IsUpper 方法的语法中各参数的说明

参　数	描　述
s	一个字符串
Index	s 中的字符位置
返回值	如果 s 中位于 index 处的字符是大写字母，则为 true；否则，为 false

ImageFormat 类

ImageFormat 类属于 System.Drawing.Imaging 命名空间，此类用于指定图像的格式，并且是不可继承的。

属性

ImageFormat 类的常用属性及说明如表 I.2 所示。

表 I.2 ImageFormat 类常用属性及说明

属　性	说　明
Bmp	获取位图图像格式（BMP）
Emf	获取增强型 Windows 图元文件图像格式（EMF）
Exif	获取可交换图像文件（Exif）格式
Gif	获取图形交换格式（GIF）图像格式
Icon	获取 Windows 图标图像格式
Jpeg	获取联合图像专家组（JPEG）图像格式
MemoryBmp	获取内存位图图像格式
Png	获取 W3C 可移植网络图形（PNG）图像格式
Tiff	获取标签图像文件格式（TIFF）图像格式
Wmf	获取 Windows 图元文件（WMF）图像格式

更多资源请访问稀酷客(www.ckook.com)

IsLeapYear 方法

DateTime 类的 IsLeapYear 方法判断年份是否为闰年。IsLeapYear 方法返回指定的年份是否为闰年的指示。

语法

```
public static bool IsLeapYear (int year)
```

IsLeapYear 方法的语法中各参数的说明如表 I.3 所示。

表 I.3 IsLeapYear 方法的语法中各参数的说明

参　数	描　述
year	表示 4 位数年份
返回值	如果 year 为闰年，为 True；否则为 False

Image 属性

Image 属性用于获取或设置显示在控件上的图像。

语法

```
[LocalizableAttribute(true)]
public Image Image { get; set; }
```

属性值：控件上显示的 Image。

ImageList 组件

Windows 窗体 ImageList 组件用于存储图像，这些图像随后可由控件显示。图像列表能够为一致的单个图像目录编写代码。例如，只需更改按钮的 ImageIndex 或 ImageKey 属性，就可旋转 Button 控件显示的图像。还可以使同一个图像列表与多个控件相关联。例如，如果使用 ListView 控件和 TreeView 控件显示同一个文件列表，则当更改图像列表中某个文件的图标时，新图标将同时显示在两个视图中。

可以将图像列表用于任何具有 ImageList 属性的控件，或用于具有 SmallImageList 和 LargeImageList 属性的 ListView 控件。可与图像列表关联的控件包括：ListView、TreeView、ToolBar、TabControl、Button、CheckBox、RadioButton 和 Label 控件。若要使图像列表与一个控件关联，请将该控件的 ImageList 属性设置为 ImageList 组件的名称。

更多资源请访问稀酷客(www.ckook.com)

属性

ImageList 组件的常用属性及说明如表 I.4 所示。

表 I.4 ImageList 组件常用属性及说明

属　性	说　明
ColorDepth	获取图像列表的颜色深度
Images	获取此图像列表的 ImageList.ImageCollection
ImageSize	获取或设置图像列表中的图像大小
ImageStream	获取与此图像列表关联的 ImageListStreamer 句柄
TransparentColor	获取或设置被视为透明的颜色

（1）Images 属性

它包含关联的控件将要使用的图片。每个单独的图像可通过其索引值或其键值来访问。

语法：

```
public ImageCollection Images { get; }
```

属性值：图像集合。

（2）ColorDepth 属性

ColorDepth 属性确定呈现图像时所使用的颜色数量。

语法：

```
public ColorDepth ColorDepth { get; set; }
```

属性值：图像的可用颜色数。

（3）ImageSize 属性

ImageSize 属性用于设置图像大小，较大的图像将缩小至适当的尺寸。

语法：

```
[LocalizableAttribute(true)]
public Size ImageSize { get; set; }
```

属性值：定义列表中的图像高度和宽度（以像素为单位）的 Size。默认的大小是 16×16。最大的大小是 256 × 256。

IndexOf 方法

在字符串中获得某数字位置时，可以使用 string 类的 IndexOf 方法，该方法用来确定指定字符在字符串中的索引，如果在字符串中能找到指定字符，则返回其索引，否则返回–1。

更多资源请访问稀酷客(www.ckook.com)

语法

```
int IndexOf(char value)
int IndexOf(char value,int startIndex)
int IndexOf(char value,int startIndex.,int count)
int IndexOf(string value)
int IndexOf(string value,int startIndex)
int IndexOf(string value,int startIndex,int count)
```

IndexOf 方法的参数说明如表 I.5 所示。

表 I.5 IndexOf 方法的参数说明

参　数	说　明
value	待定位的字符或者子串
startIndex	在总串中开始搜索的起始位置
count	在总串中从起始位置开始搜索的字符数

InitialDirectory 属性

InitialDirectory 属性用于获取或设置文件对话框显示的初始目录。

语法

```
public string InitialDirectory { get; set; }
```

属性值：文件对话框中显示的初始目录。默认值为空字符串 ("")。

Insert 方法

开发程序时，有时会根据需要在字符串中添加一些空格，这时可以使用 string 类的 Insert 方法 ，该方法可以在字符串中的指定位置插入一个新的字符串（包括空格）。

语法

```
public string Insert (int startIndex, string value);
```

Insert 方法的语法中各参数的说明如表 I.6 所示。

表 I.6 Insert 方法的语法中各参数的说明

参　数	描　述
startIndex	用于指定所要插入的位置，索引从 0 开始
value	指定所要插入的字符串
返回值	指定字符串的一个新 String 等效项，但在位置 startIndex 处插入 value

更多资源请访问稀酷客(www.ckook.com)

Interval 属性

Interval 属性用于获取或设置引发 Elapsed 事件的间隔。



语法

```
public double Interval { get; set; }
```

属性值：引发 Elapsed 事件的间隔时间（以毫秒为单位）。默认为 100 毫秒。



示例

本示例将 Timer 组件的 Interval 属性设置为 1000。

```
this.timer1.Interval = 1000;
```

本示例设置 Timer 控件的间隔时间和停止控件。

```
this.timer1.Interval = 1000;
this.timer1.Enabled = false;
```

IPEndPoint 类

IPEndPoint 类将网络端点表示为 IP 地址和端口号。IPEndPoint 类包含应用程序连接到主机上的服务所需的主机和本地或远程端口信息。通过组合服务的主机 IP 地址和端口号，IPEndPoint 类形成到服务的连接点。



属性

IPEndPoint 类的常用属性及说明如表 I.7 所示。

表 I.7 IPEndPoint 类的常用属性及说明

属　性	说　明
Address	获取或设置终结点的 IP 地址
AddressFamily	获取网际协议（IP）地址族
Port	获取或设置终结点的端口号

（1）Address 属性

获取或设置终结点的 IP 地址。

语法：

```
public IPAddress Address { get; set; }
```

属性值：包含终结点的 IP 地址的 IPAddress 实例。

更多资源请访问稀酷客(www.ckook.com)

（2）AddressFamily 属性

获取网际协议（IP）地址族。

语法：

```
public override AddressFamily AddressFamily { get; }
```

属性值：返回 AddressFamily.InterNetwork。

（3）Port 属性

获取或设置终结点的端口号。

语法：

```
public int Port { get; set; }
```

属性值：介于 MinPort 到 MaxPort 之间的整数值，指示终结点的端口号。

方法

IPEndPoint 类的常用方法及说明如表 I.8 所示。

表 I.8 IPEndPoint 类的常用方法及说明

方　法	说　明
Create	从套接字地址创建终结点
Serialize	将终结点信息序列化为 SocketAddress 实例

（1）Create 方法

从套接字地址创建终结点。

语法：

```
public override EndPoint Create (SocketAddress socketAddress)
```

Create 方法的语法中各参数的说明如表 I.9 所示。

表 I.9 Create 方法的语法中各参数的说明

参　数	描　述
socketAddress	用于终结点的 SocketAddress
返回值	使用指定套接字地址的 EndPoint 实例

（2）Serialize 方法

将终结点信息序列化为 SocketAddress 实例。

语法：

```
public override SocketAddress Serialize ()
```

返回值：一个 SocketAddress 实例，包含终结点的套接字地址。

更多资源请访问稀酷客(www.ckook.com)

IPHostEntry 类

为 Internet 主机地址信息提供容器类。IPHostEntry 类将一个域名系统(DNS) 主机名与一组别名和一组匹配的 IP 地址关联。IPHostEntry 类作为 Helper 类和 Dns 类一起使用。

属性

IPHostEntry 类的常用属性及说明如表 I.10 所示。

表 I.10 IPHostEntry 类的常用属性及说明

属 性	说 明
AddressList	获取或设置与主机关联的 IP 地址列表
Aliases	获取或设置与主机关联的别名列表
HostName	获取或设置主机的 DNS 名称

(1) AddressList 属性

获取或设置与主机关联的 IP 地址列表。

语法：

```
public IPAddress[] AddressList { get; set; }
```

属性值：一个 IPAddress 类型的数组，包含解析为 Aliases 属性中包含的主机名的 IP 地址。

(2) Aliases 属性

获取或设置与主机关联的别名列表。

语法：

```
public string[] Aliases { get; set; }
```

属性值：一组字符串，包含解析为 AddressList 属性中的 IP 地址的 DNS 名称。

(3) HostName 属性

获取或设置主机的 DNS 名称。

语法：

```
public string HostName { get; set; }
```

属性值：包含服务器的主要主机名的字符串。

IsMatch 方法

在指定的输入字符串中搜索 Regex 构造函数中指定的正则表达式匹配项。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public Match Match (string input)
```

IsMatch 方法的语法中各参数的说明如表 I.11 所示。

表 I.11 IsMatch 方法的语法中各参数的说明

参　数	描　述
input	要搜索匹配项的字符串
返回值	一个正则表达式 Match 对象

IsMdiContainer 属性

获取或设置一个值，该值指示窗体是否为多文档界面（MDI）子窗体的容器。此属性将窗体的显示和行为更改为 MDI 父窗体。当此属性设置为 true 时，该窗体显示具有凸起边框的凹陷工作区。所有分配给该父窗体的 MDI 子窗体都在该父窗体的工作区内显示。

语法

```
public bool IsMdiContainer { get; set; }
```

属性值：如果该窗体是 MDI 子窗体的容器，则为 true；否则为 false。默认为 false。

Items 属性

获取或设置数组列表对象中的项的集合，默认值为 Empty。

语法

```
public ObjectCollection Items { get; }
```

属性值：数组列表对象中的项的集合。

更多资源请访问稀酷客(www.ckook.com)

J

Join（String 类）方法

在指定 String 数组的每个元素之间串联指定的分隔符 String，从而产生单个串联的字符串。

语法

```
public static string Join (string separator,string[] value)
```

Join 方法的语法中各参数的说明如表 J.1 所示。

表 J.1　Join 方法的语法中各参数的说明

参　数	描　述
separator	字符串
value	一个字符串数组
返回值	包含与 separator 字符串交错的 value 的元素

Join（Thread 类）方法

在继续执行标准的 COM 和 SendMessage 消息泵处理期间，阻止调用线程，直到某个线程终止为止。

语法

```
public void Join ()
```

使用此方法确保线程已终止。如果线程不终止，则调用方将无限期阻止。

更多资源请访问稀酷客(www.ckook.com)

K

KeyChar 属性

KeyChar 属性获取或设置与按下的键对应的字符。

语法

```
public char KeyChar { get; set; }
```

属性值：键盘键所对应的 ASCII 字符。

示例

本示例判断用户是否按下回车键。

```
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 13)
    { MessageBox.Show("您按下回车键了"); }
}
```

本示例在窗体的 KeyPress 事件中，通过 KeyChar 属性判断用户是否按下回车键。

```
private void Form1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 13)
    {
        this.Text = "按下回车键";
    }
}
```

KeyDown 事件

在控件有焦点的情况下，按下键时发生。

语法

```
public event KeyEventHandler KeyDown
```


更多资源请访问稀酷客(www.ckook.com)

KeyPress 事件

此事件在控件有焦点的情况下，按下键时发生。



语法

```
public event KeyPressEventHandler KeyPress
```

KeyUp 事件

此事件在控件有焦点的情况下，释放键时发生。



语法

```
public event KeyEventHandler KeyUp
```



示例

本示例在 KeyUp 事件下，向用户显示一个“帮助”弹出窗口。

```
    private void textBox1_KeyUp(object sender, System.Windows.Forms.
KeyEventArgs e)
    {
        if (e.KeyCode == Keys.F1)
        {
            Help.ShowPopup(textBox1, "请正确输入密码", new Point(textBox1.
Right, this.textBox1.Bottom));
        }
    }
```

本示例按完键盘后，自动设置窗体的大小。

```
    private void Form1_KeyUp(object sender, KeyEventArgs e)
    {
        this.ClientSize = new System.Drawing.Size(800,100);
    }
```

Kill 方法

Kill 方法强制关闭一个进程，将引起进程的非正常退出，丢失进程所有的资源或数据，而 CloseMainWindow 是请求关闭进程。如果调用 Kill，则可能丢失进程编辑的数据或分配给进程的资源。Kill 导致进程不正常终止，因而只应在必要时使用。CloseMainWindow 使进程能够有序终止并关闭所有窗

更多资源请访问稀酷客(www.ckook.com)

口，所以对于有界面的应用程序，使用它更好。如果 CloseMainWindow 失败，则可以使用 Kill 终止进程。Kill 是终止没有图形化界面的进程的唯一方法。



语法

```
public void Kill ()
```


更多资源请访问稀酷客(www.ckook.com)

Label 控件

Label 控件即标签控件，它是 Windows 应用程序中最基本的控件。Label 控件用来显示用户不能直接改变的文本信息。可以在“属性”窗口中设置控件的文本信息，也可以通过编写代码来设置要显示的文本信息。向窗体添加 Label 控件时，系统会自动创建 Label 控件的一个对象。

属性

Label 控件常用属性及说明如表 L.1 示。

表 L.1 Label 控件常用属性及说明

属　性	说　明
Image	此属性用于指定将在标签上显示的图像
Text	此属性是与文件相关联的文本
Name	指示代码中用来标识该对象的名称
Enabled	指示是否已启用该控件
Visible	确定该控件是可见的还是隐藏的

下面对比较重要的属性进行详细介绍。

（1）Text 属性

用来设置要在该控件上显示的文本。

语法：

```
public override string Text { get; set; }
```

例如，在 Label 控件中显示“欢迎你”文本信息，程序主要代码如下：

```
private void frmLabel_Load(object sender, EventArgs e)
{
    this.label1.Text = "欢迎你";
}
```

（2）Visible 属性

用于设置控件是否可见。

语法：

```
public bool Visible { get; set; }
```

属性值：如果显示该控件，则为 true，否则为 false。默认为 true。

更多资源请访问稀酷客(www.ckook.com)

例如，下面的代码用来设置 Label2 控件可见：

```
label2.Visible = true;
```



方法

Label 控件常用方法及说明如表 L.2 所示。

表 L.2　Label 控件常用方法及说明

方　法	说　明
Select	激活控件
Show	该方法将控件的 Visible 属性设为 true 并显示控件
Hide	该方法用于隐藏控件

下面对比较重要的方法进行详细介绍。

（1）Show 方法

该方法将控件的 Visible 属性设为 true 并显示控件。

语法：

```
public void Show ()
```

（2）Hide 方法

该方法用于隐藏控件。

语法：

```
public void Hide ()
```

例如，将 label2 控件隐藏，代码如下：

```
label2.Hide();
```



事件

Label 控件常用事件及说明如表 L.3 所示。

表 L.3　Label 控件常用事件及说明

事　件	说　明
KeyPress	当标签有焦点且用户按下某个键时发生
KeyUp	当标签有焦点且用户释放某个键时发生
Click	当单击标签时发生的触发
Paint	此事件在重绘或更新控件时发生

LabelEdit 属性

ListView 控件的 LabelEdit 属性允许用户编辑 ListView 控件数据列中的内

更多资源请访问稀酷客(www.ckook.com)

容。默认 ListView 控件是只读的，用户只能查看内容不能修改，如果想修改设置 LabelEdit 属性即可。当 LabelEdit 属性设置为 True 时，用户可以修改项文本：单击项文本，将其选中，然后再次单击项文本，将标签文本置于编辑模式，就可以修改或替换项的文本标签了。在用户编辑项文本之前和之后，可以使用 BeforeLabelEdit 和 AfterLabelEdit 事件来执行任务。如果该属性设置为 True，则无法修改子项的文本。要更改子项的文本，可以在用户双击控件中的子项时为其显示对话框。



语法

```
public bool LabelEdit { get; set; }
```

属性值：如果用户可以在运行时编辑项的标签，则为 true；否则，为 false。默认为 false。

Last 函数

Last 函数用于返回在查询所返回的结果集中的最后一个记录的字段值。



语法

```
LAST(expression)
```

Last 函数的语法中各参数的说明如表 L.4 所示。

表 L.4 Last 函数的语法中各参数的说明

参　数	描　述
expression	代表一个字符串表达式，它标识了包含要使用数据的字段，或者是使用该字段中数据执行计算的表达式。expression 中的操作数可以包括表字段、常量或函数（可以是固有的或者用户自定义的函数，但不能是其他 SQL 聚合函数）的名称

LastIndexOf 方法

获得某字符在字符串中最后出现的位置时，可以使用 string 类的 LastIndexOf 方法，该方法用来确定指定字符在字符串中最后一次出现的索引位置，如果在字符串中找到指定字符，则返回其索引，否则返回–1。



语法

```
public int LastIndexOf (string value)
```

LastIndexOf 方法的语法中各参数的说明如表 L.5 所示。

更多资源请访问稀酷客(www.ckook.com)

表 L.5 LastIndexOf 方法的语法中各参数的说明

参　数	描　述
value	要查找的 String
返回值	如果找到该字符，则为 value 的索引位置；如果未找到该字符，则为-1。如果 value 为 Empty，则返回值为 value 中的最后一个索引位置

LayoutMdi 方法

在 MDI 父窗体内排列多文档界面（MDI）子窗体。



语法

```
public void LayoutMdi (MdiLayout value)
```

LayoutMdi 方法的语法中各参数的说明如表 L.6 所示。

表 L.6 LayoutMdi 方法的语法中各参数的说明

参　数	描　述
value	MdiLayout 值之一，定义 MDI 子窗体的布局

Leave 事件

Leave 事件在输入焦点离开控件时发生。



语法

```
public event EventHandler Leave
```

Lines 属性

获取或设置多行配置中的文本行。



语法

```
public string[] Lines { get; set; }
```

属性值：只包含单行的 String 类型数组。



示例

本示例使用 Lines 属性来实现设置 TextBox 控件输入多行文本。

更多资源请访问稀酷客(www.ckook.com)

```
private void button1_Click(object sender, EventArgs e)
{
    string[] strAddress = { "one", "two", "three","four","five" };
    textBox1.Multiline = true;
    textBox1.ScrollBars = ScrollBars.Both;
    textBox1.Lines=strAddress;
}
```



注意

Lines属性是从TextBoxBase基类继承的。但是，由于MaskedTextBox控件不支持多行配置，因此该属性不受支持。

LinkLabel 控件

LinkLabel 控件使用户能够向 Windows 窗体应用程序添加 Web 样式的链接。一切可以使用 Label 控件的地方都可以使用 LinkLabel 控件，另外，用户可以将文本的一部分设置为指向某个对象或网页的链接。

属性

LinkLabel 控件常用属性及说明如表 L.7 所示。

表 L.7 LinkLabel 控件常用属性及说明

属　性	说　明
ActiveLinkColor	用于指定表示用户单击超级链接时超链接显示的颜色
DisabledLinkColor	用于指定表示超级链接被禁止用时的颜色
LinkArea	用于表示必须显示为超级链接的文本
LinkBehaviour	用于表示链接行为。它可用于指定超级链接是带下画线等
LinkColor	用于表示超级链接初始的颜色
VisitedLinkcolor	用于表示超级链接以前访问过的颜色
Dock	用于表示超级链接在容器的布局

下面详细介绍 LinkArea 属性，该属性用于表示必须显示为超级链接的文本。

```
public LinkArea LinkArea { get; set; }
```

该属性为可读可写属性。

例如，将“窗体”两个字设为超级链接文本，代码如下：

```
linkLabel1.LinkArea = new LinkArea(6, 8);
```


更多资源请访问稀酷客(www.ckook.com)

方法

LinkLabel 控件常用方法及说明如表 L.8 所示。

表 L.8 LinkLabel 控件常用方法及说明

方 法	说 明
Select	激活控件
Show	指示是否已启用该控件
Hide	确定该控件是可见的还是隐藏的

事件

LinkLabel 控件常用事件及说明如表 L.9 所示。

表 L.9 LinkLabel 控件常用事件及说明

事 件	说 明
DoubleClick	在双击控件时发生
Enter	进入控件时发生
LinkClicked	当单击控件内的链接时发生

ListBox 控件

ListBox 控件显示较长的选项列表，用户可从中选择一项或多项。如果项总数超出可以显示的项数，则自动向 ListBox 控件添加滚动条。ListBox 控件列表中的每个元素称为项。

属性

ListBox 控件常用属性及说明如表 L.10 所示。

表 L.10 ListBox 控件常用属性及说明

属 性	说 明
Items	此属性用于查看列表框中的项
MultiColumn	属性设置为 true 时列表框以多列形式显示项，并且会出现一个水平滚动条。属性设置为 false 时，列表框以单列形式显示项，并且会出现一个垂直滚动条
ScrollAlwaysVisible	属性设置为 true 无论项数多少都将显示滚动条
SelectionMode	属性确定一次可以选择多少列表项
SelectedIndex	此属性代表当前选中项的索引号，如果未选定任何项，则 SelectedIndex 值为–1。如果选定了列表中的第一项，则 SelectedIndex 值为 0。当选定多项时，SelectedIndex 值反映列表中最先出现的选定项

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	说　明
Count	属性反映列表的项数，由于 SelectedIndex 是从 0 开始的，所以 Count 属性的值通常比 SelectedIndex 的最大可能值大 1
SelectedItem	此属性表示当前选中的项
SelectedItems	此属性表示当前选中的所有项
SelectedValue	此属性表示当前选中项的值
sorted	此属性的值确定是否对列表框中的项进行排序
Text	此属性表示当前选中的项的文本

下面对比较重要的属性进行详细介绍。

（1）Items 属性

该属性用于查看列表框中的项。

语法：

```
public ObjectCollection Items { get; }
```

属性值：ListBox.ObjectCollection 表示 ListBox 中的项。



注意

（1）该属性使用户可以获取对当前存储在 ListBox 中的项列表的引用。通过此引用，可以在集合中添加项、移除项和获得项的计数。

（2）可以使用 DataSource 属性来操控 ListBox 的项。如果使用 DataSource 属性向 ListBox 添加项，则可以使用 Items 属性查看 ListBox 中的项，但不能使用 ListBox.ObjectCollection 的方法向该列表添加项或从中移除项。

（2）SelectedItem 属性

表示当前选中的项。

语法：

```
public Object SelectedItem { get; set; }
```

属性值：表示控件中当前选定内容的对象。



注意

对于标准 ListBox，可以使用此属性确定在 ListBox 中选定了哪个项。如果 ListBox 的 SelectionMode 属性设置为 SelectionMode.MultiSimple 或 SelectionMode.MultiExtended（它指示多重选择 ListBox），并在该列表中选定了多个项，则此属性可返回任何选定的项。

更多资源请访问稀酷客(www.ckook.com)

方法

ListBox 控件常用方法及说明如表 L.11 所示。

表 L.11 ListBox 控件常用方法及说明

方　法	说　明
FindString	从列表中检索以指定字符串开始的第一项
GetSelected	如果选中了指定项，此方法返回 true，如果没有选中指定项，则返回 false
SetSelected	选择或取消选定的项。可将一个 Boolean 值 ture 传递至此方法，以选该项

下面详细介绍 FindString 方法，该方法用于从列表中检索以指定字符串开始的第一项。

语法：

```
public int FindString (string s)
public int FindString (string s,int startIndex)
```

FindString 方法的语法中各参数的说明如表 L.12 所示。

表 L.12 FindString 方法的语法中各参数的说明

参　数	描　述
s	要搜索的文本
startIndex	项的从零开始的索引，该项在要搜索的第一个项之前。设置为-1，以从控件的开始处进行搜索
返回值	找到的第一个项的从零开始的索引；如果找不到匹配项，则返回 ListBox.NoMatches

事件

ListBox 控件常用事件及说明如表 L.13 所示。

表 L.13 ListBox 控件常用事件及说明

事　件	说　明
DataSourceChanged	当 DataSource 更改时发生
SelectedIndexChanged	当 SelectedIndex 属性值发生更改时，触发此事件
SelectedValueChanged	当 SelectedValue 属性值发生更改时，触发此事件

ListView 控件

ListView 控件可以显示带图标的项列表，用户可使用该控件创建类似 Windows 资源管理器那样的用户界面。ListView 控件具有 4 种视图模式：

（1）仅文本，这是默认视图，此视图下，只显示列表项的文本。

更多资源请访问稀酷客(www.ckook.com)

（2）带有小图标的文本，此视图下，小图标随列表项的文本同时显示。

（3）带有大图标的文本，此视图下，大图标随列表项的文本同时显示。

（4）报表视图，此视图下，列表项显示在多个列中。

属性

ListView 控件常用属性及说明如表 L.14 所示。

表 L.14　ListView 控件常用属性及说明

属　性	说　明
FullrowSelect	属性用于指定在 ListView 控件中单击某项时要执行的操作过程。单击某项时，可以指定是只应该选择该项，还是应选该项所在的整行
View	此属性将指定将建列表视图的类型。视图类型主要包括：大图标、小图标、列表和详细信息
Alignment	指定 ListView 各项的对齐方式
Sorting	对项进行排序的方式
Multiselect	此属性设为 true 时，表示在控件中一次可以选择多个项
GridLines	获取或设置一个值，该值指示，在包含控件中项及其子项的行和列之间是否显示网格线

方法

ListView 控件常用方法及说明如表 L.15 所示。

表 L.15　ListView 控件常用方法及说明

方　法	说　明
HitTest	该方法是指在给定 x 和 y 坐标的情况下，提供项信息
Clear	该方法用于删除 ListView 控件中所有的项
ResetText	将 Text 属性重置为其默认值

事件

ListView 控件常用事件及说明如表 L.16 所示。

表 L.16　ListView 控件常用事件及说明

事　件	说　明
DoubleClick	在双击控件时发生
ItemCheck	该事件在选中 ListView 控件项时触发此事件
Click	该事件在单击 ListView 控件列时触发此事件

更多资源请访问稀酷客(www.ckook.com)

Load 方法

Load 方法从指定的 XmlReader 加载 XML 文档。

语法

```
public virtual void Load (XmlReader reader)
```

Load 方法的语法中各参数的说明如表 L.17 所示。

表 L.17 Load 方法的语法中各参数的说明

参　数	描　述
reader	用于将 XML 数据输送到文档中的 XmlReader

LoadFile 方法

LoadFile 方法用于指定文件的内容加载或检索到 RichTextBox 控件中，当使用 LoadFile 方法加载文件时，所加载文件的内容将替换 RichTextBox 控件的整个内容。这将导致 Text 和 Rtf 属性的值更改。可以使用此方法将以前创建的文本或 RTF 文档加载到控件中以进行操作。如果想保存文件，可以使用 SaveFile 方法。

语法

```
public void LoadFile (Stream data,RichTextBoxStreamType fileType)
public void LoadFile (string path,RichTextBoxStreamType fileType)
public void LoadFile (string path)
```

LoadFile 方法的语法中各参数的说明如表 L.18 所示。

表 L.18 LoadFile 方法的语法中各参数的说明

参　数	描　述
path	要加载到控件中的文件的名称和位置
fileType	RichTextBoxStreamType 值之一
data	要加载到 RichTextBox 控件中的数据流

示例

本示例通过 LoadFile 方法将选择的文本文件加载到 RichTextBox 控件中显示出来，如图 L.1 所示。

更多资源请访问稀酷客(www.ckook.com)

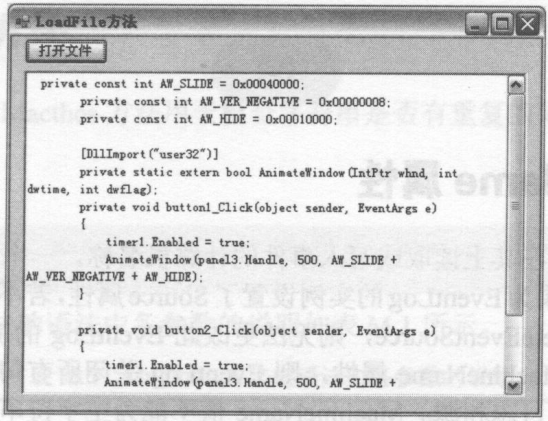


图 L.1　通过 LoadFile 方法加载数据

主要代码如下：

```
    private void button1_Click(object sender, EventArgs e)
    {
        if (openFileDialog1.ShowDialog() == DialogResult.OK)
        {
    richTextBox1.LoadFile(openFileDialog1.FileName,RichTextBoxStream
Type.PlainText);
        }
    }
```



注意

使用此版本的 LoadFile 方法时，如果所加载的文件不是 RTF 文档，则将出现异常。若要加载不同类型的文件（如 ASCII 文本文件），请使用接受 RichTextBoxStreamType 枚举值作为参数的此方法的其他版本。

Location 属性

Location 属性用于获取或设置该控件的左上角相对于其容器的左上角的坐标。



语法

```
    [LocalizableAttribute(true)]
    public Point Location { get; set; }
```

属性值：Point，它表示控件的左上角相对于其容器的左上角。

更多资源请访问稀酷客(www.ckook.com)

M

MachineName 属性

获取或设置在其上读取或写入事件的计算机名称。一个源一次只能注册到一个日志。如果为 EventLog 的实例设置了 Source 属性，若不事先更改 Source 的值或调用 DeleteEventSource，则无法更改此 EventLog 的 MachineName 属性。如果更改 MachineName 属性，则 EventLog 关闭所有句柄，并重新附加到新计算机上的日志和源。MachineName 值不能为空字符串。如果未显式设置该值，它将默认为本地计算机（"."）。

语法

```
public string MachineName { get; set; }
```

属性值：正在运行与此 ServiceController 实例关联的服务的计算机名称。默认为本地计算机（“.”）。

示例

本示例判断 IIS 服务状态，其中设置 ServiceController 组件的 MachineName 属性为“.”。

```
private void button3_Click(object sender, EventArgs e)
{
    serviceController1.MachineName = ".";0
    serviceController1.ServiceName = "IISAdmin";//IIS 服务
    if (serviceController1.Status==ServiceControllerStatus.Stopped)
    {
        MessageBox.Show(serviceController1.DisplayName + "服务已停止");
        button1.Enabled = true;
    }
    else
    {
      MessageBox.Show(serviceController1.DisplayName + "  服务已开启");
      button2.Enabled = true;
    }
}
```


更多资源请访问稀酷客(www.ckook.com)

Macthes 方法

Regex 类的 Macthes 方法用于检查字符串是否有重复的词出现。



语法

```
public static MatchCollection Matches (string input,string
pattern,RegexOptions options)
```

Macthes 方法的语法中各参数的说明如表 M.1 所示。

表 M.1　Macthes 方法的语法中各参数的说明

参　数	描　述
input	要搜索匹配项的字符串
pattern	要匹配的正则表达式模式
options	RegexOption 枚举值的按位“或”组合
返回值	搜索找到的 Match 对象的 MatchCollection

MatchCollection 对象表示通过以迭代方式将正则表达式模式应用于输入字符串所找到的成功匹配的集合。RegexOption 枚举值及说明如表 M.2 所示。

表 M.2　RegexOption 枚举值及说明

枚举值	说　明
Compiled	指定将正则表达式编译为程序集。这会产生更快的执行速度，但会增加启动时间
CultureInvariant	指定忽略语言中的区域性差异。有关更多信息可以参见在 Regular Expressions 命名空间中执行不区分区域性的操作
ECMAScript	为表达式启用符合 ECMAScript 的行为。该值只能与 IgnoreCase、Multiline 和 Compiled 值一起使用，与其他任何值一起使用均将导致异常
ExplicitCapture	指定有效的捕获仅为形式（?<name>...）的显式命名或编号的组。这使未命名的圆括号可以充当非捕获组，并且不会使表达式的语法（?…）显得笨拙
IgnoreCase	指定不区分大小写的匹配
IgnorePatternWhitespace	消除模式中的非转义空白并启用由“#”标记的注释。但是，IgnorePatternWhitespace 值不会影响或消除字符类中的空白
Multiline	多行模式。更改“^”和“$”的含义，使它们分别在任意一行的行首和行尾匹配，而不仅仅在整个字符串的开头和结尾匹配
None	指定不设置选项
RightToLeft	指定搜索从右向左而不是从左向右进行
Singleline	指定单行模式。更改点（.）的含义，使它与每一个字符匹配（而不是与除\n 之外的每个字符匹配）

更多资源请访问稀酷客(www.ckook.com)

MailMessage 类

MailMessage 类主要用于指定邮件的发送地址、收件人地址等。

属性

MailMessage 类的常用属性及说明如表 M.3 所示。

表 M.3 MailMessage 类的常用属性及说明

属　性	说　明
Attachments	获取用于存储附加到此电子邮件的数据的附件集合
Body	获取或设置邮件正文
From	获取或设置此电子邮件的发信人地址
Sender	获取或设置此电子邮件的发件人地址
Subject	获取或设置此电子邮件的主题
To	获取包含此电子邮件的收件人的地址集合

示例

自定义一个 CreateTimeoutTestMessage 方法，该方法中通过 MailMessage 类与 SmtpClient 类实例化 Mail 对象与 Client 对象，然后使用 Mail 对象来完成对邮件信息的定制，如收件人、主题和内容等，最后调用 Client 对象的 Send 方法来完成信息的发送功能。CreateTimeoutTestMessage 方法关键代码如下：

```
public  void CreateTimeoutTestMessage(string server)
{
    try
    {
        Mail = new MailMessage(this.txtFrom.Text, this.txtGet.Text);
        Mail.Subject =this.txtSubject.Text.Trim().ToString();
        Mail.Body = this.richTextBox1.Text.Trim().ToString();
        Client = new SmtpClient(server, 25);
        Client.Send(Mail);
        MessageBox.Show("Success!!! ");
    }
    catch (Exception ey)
    {
        MessageBox.Show(ey.Message);
    }
}
```


更多资源请访问稀酷客(www.ckook.com)

ManagementClass 类

表示公共信息模型管理类。管理类是一个 WMI 类，如 Win32_LogicalDisk 和 Win32_Process，前者表示磁盘驱动器，后者表示进程（如 Notepad.exe）。



属性

ManagementClass 类的常见属性及说明如表 M.4 所示。

表 M.4　ManagementClass 类的常见属性及说明

属　性	说　明
Derivation	获取一个数组，该数组包含继承层次结构中从该类到层次结构顶部的所有 WMI 类
Methods	获取或设置 MethodData 对象的集合，这些对象表示 WMI 类中定义的方法
Path	获取或设置 ManagementClass 对象绑定到的 WMI 类的路径

（1）Derivation 属性

获取一个数组，该数组包含继承层次结构中从该类到层次结构顶部的所有 WMI 类。

语法：

```
public StringCollection Derivation { get; }
```

属性值：一个字符串集合，它包含该类的继承层次结构中所有 WMI 类的名称。

（2）Methods 属性

获取或设置 MethodData 对象的集合，这些对象表示 WMI 类中定义的方法。

语法：

```
public MethodDataCollection Methods { get; }
```

属性值：表示 WMI 类中定义的方法的 MethodDataCollection。

（3）Path 属性

获取或设置 ManagementClass 对象绑定到的 WMI 类的路径。

语法：

```
public override ManagementPath Path { get; set; }
```

属性值：对象的类的路径。



方法

ManagementClass 类的常用方法及说明如表 M.5 所示。

更多资源请访问稀酷客(www.ckook.com)

表 M.5 ManagementClass 类的常用方法及说明

方　法	说　明
CreateInstance	初始化 WMI 类的新实例
Derive	从此类派生新类
GetInstances	返回该类的所有实例的集合
GetRelatedClasses	检索与 WMI 类相关的类
GetRelationshipClasses	检索使此类与其他类相关的关系类
GetStronglyTypedClassCode	为给定的 WMI 类生成强类型类
GetSubclasses	返回该类的所有派生类的集合

ManagementObject 类

ManagementObject 表示 WMI 实例。

属性

ManagementObject 类的常用属性及说明如表 M.6 所示。

表 M.6 ManagementObject 类的常用属性及说明

属　性	说　明
ClassPath	获取或设置对象的类的路径
Item	通过“[]”符号获取对属性值的访问，此属性是 ManagementBaseObject 类的索引器。用户可以使用由某个类型定义的默认索引属性，但不能显式定义自己的属性，但是，如果在某个类上指定 expando 属性，将自动提供一个类型为 Object、索引类型为 String 的默认索引属性
Options	获取或设置检索对象时要使用的其他信息
Path	获取或设置对象的 WMI 路径
Properties	获取描述管理对象属性的 PropertyData 对象的集合
Qualifiers	获取管理对象中定义的 QualifierData 对象的集合。集合中的每个元素均包含限定符名称、值和风格等信息
Scope	获取或设置此对象在其中驻留的范围
SystemProperties	获取管理对象的 WMI 系统属性的集合（例如，类名、服务器和命名空间等）。WMI 系统属性名以“__”开头。

方法

ManagementObject 类的常用方法及说明如表 M.7 所示。

更多资源请访问稀酷客(www.ckook.com)

表 M.7　ManagementObject 类的常用方法及说明

方　法	说　明
CopyTo	将对象复制到另一个位置
Delete	删除对象
Get	绑定到管理对象
GetPropertyQualifierValue	返回指定的属性限定符的值
GetPropertyValue	获取某属性值的等效访问器
GetQualifierValue	获取指定的限定符的值
GetText	以指定的格式返回对象的文本化表示形式
op_Explicit	提供由 ManagementObject 表示的内部 WMI 对象
Put	提交对对象所做的更改
SetPropertyQualifierValue	设置指定的属性限定符的值
SetPropertyValue	设置指定属性的值
SetQualifierValue	设置指定的限定符的值

MaskedTextBox 控件

MaskedTextBox 控件使用掩码区分正确和不正确的用户输入。



属性

MaskedTextBox 控件常用属性及说明如表 M.8 所示。

表 M.8　MaskedTextBox 控件常用属性及说明

属　性	说　明
PromptChar	此属性获取或设置用于表示 MaskedTextBox 中缺少用户输入的字符
RejectInputOnFirstFailure	此属性获取或设置一个值，该值指示当遇到第一个无效字符时，是否应停止对用户输入的分析
ResetOnPrompt	此属性获取或设置一个值，该值确定应如何处理与提示字符匹配的输入字符
ResetOnSpace	此属性获取或设置一个值，该值确定应如何处理空白输入字符
TextMaskFormat	此属性获取或设置一个值，该值决定原义字符和提示字符是否包括在格式化字符串中
UseSystemPasswordChar	此属性获取或设置一个值，该值指示是否应使用操作系统提供的密码字符
ValidatingType	此属性获取或设置用于验证用户输入的数据的数据类型
Mask	此属性获取或设置运行时使用的输入掩码

更多资源请访问稀酷客(www.ckook.com)

方法

MaskedTextBox 控件常用方法及说明如表 M.9 所示。

表 M.9 MaskedTextBox 控件常用方法及说明

方 法	说 明
Copy	将文本框中的当前选定内容复制到“剪贴板”
ValidateText	将用户输入字符串转换为验证类型的一个实例
DrawToBitmap	支持呈现到指定的位图

事件

MaskedTextBox 控件常用事件及说明如表 M.10 所示。

表 M.10 MaskedTextBox 控件常用事件及说明

事 件	说 明
MaskInputRejected	当用户的输入或者分配的字符与输入掩码的对应格式元素不匹配时发生
MaskChanged	在输入掩码更改后发生
TextChanged	在 Text 属性值更改时发生

MAX 函数

MAX 函数返回表达式中的最大值。

语法

```
MAX ( [ ALL ] expression )
```

MAX 函数的语法中各参数的说明如表 M.11 所示。

表 M.11 MAX 函数的语法中各参数的说明

参 数	描 述
ALL	对所有值应用聚合函数，默认值为 ALL
expression	常量、列名或函数，以及算术运算符、位运算符和字符串运算符的任意组合。MAX 可以用于数字、字符和 datetime 列，但不能用于 bit 列，不允许使用聚合函数和子查询

示例

本示例使用 MAX 查询学生成绩表（student）中英语成绩（stuEnglish）最高分的学生信息。

```
select *  from student where stuEnglish=(select Max(stuEnglish) from
```


更多资源请访问稀酷客(www.ckook.com)

```
student)
```

本示例通过 Math 类中的 Max 方法比较指定两个数的大小，返回两个数中较大的数。

```
private void Form1_Load(object sender, EventArgs e)
{
    int qq= Math.Max(11,33);
}
```

MaxDate 属性

获取或设置可在控件中选择的最大日期和时间。

语法

```
public DateTime MaxDate { get; set; }
```

属性值：可在控件中选择的最大日期和时间。默认值为 12/31/9998 23:59:59。

示例

本示例创建 DateTimePicker 控件的一个新实例并将其初始化。设置控件的 CustomFormat 属性。此外，还要设置 ShowCheckBox 属性以便此控件显示 CheckBox。并设置 ShowUpDown 属性以便此控件显示为数值调节钮控件（也称为 up-down 控件）。

```
public void CreateMyDateTimePicker()
{
   DateTimePicker dateTimePicker1 = new DateTimePicker();
   dateTimePicker1.MinDate = new DateTime(1985, 6, 20);
   dateTimePicker1.MaxDate = DateTime.Today;
   dateTimePicker1.CustomFormat = "MMMM dd, yyyy - dddd";
   dateTimePicker1.Format = DateTimePickerFormat.Custom;
   dateTimePicker1.ShowCheckBox = true;
   dateTimePicker1.ShowUpDown = true;
}
```

Maximum 属性

Maximum 属性用于获取或设置数字显示框的最大值。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public decimal Maximum { get; set; }
```

属性值：数字显示框的最大值。默认值为 100。

示例

例如，下面以 ProgressBar 控件为例，通过代码用来设置 Minimum 属性和 Maximum 属性的值。

```
progressBar1.Minimum = 0;
progressBar1.Maximum = 5000;
```

本示例通过设置 NumericUpDown 控件的 Maximum 属性，设置该控件允许的最大值。

```
private void Form1_Load(object sender, EventArgs e)
{
    numericUpDown1.Maximum = 1000;
}
```

本示例通过设置 HScrollBar 控件的 Maximum 属性，设置该控件允许的最大值。

```
private void Form1_Load(object sender, EventArgs e)
{
    hScrollBar1.Maximum = 100;
}
```

本示例通过设置 TrackBar 控件的 Maximum 属性，设置该控件允许的最大值。

```
private void Form1_Load(object sender, EventArgs e)
{
    this.trackBar1.Maximum = 100;
}
```

MDI 窗体

MDI 窗体（Multiple-Document Interface），即多文档界面，它由一个父窗体和若干个子窗体组成。MDI 窗体允许同时显示多个子窗体，它常包括含有子菜单的“窗口”菜单项，该菜单项用于在窗口之间进行切换。

创建父窗体

创建 MDI 父窗体的步骤如下：

更多资源请访问稀酷客(www.ckook.com)

（1）依次选择“开始”/“所有程序”/“Microsoft Visual Studio 2005”/“Microsoft Visual Studio 2005”选项，打开“Visual Studio 2005”开发环境，在“文件”菜单中，选择“新建”/“项目”选项，如图 M.1 所示。

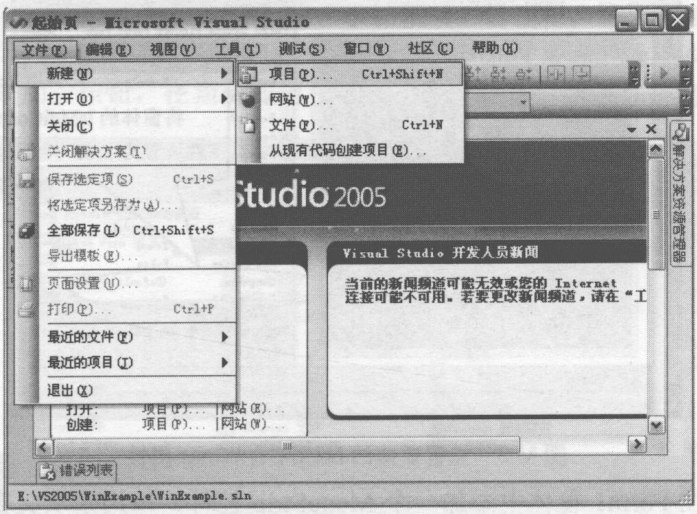


图 M.1　选择“新建”/“项目”选项

（2）弹出如图 M.2 所示的“新建项目”对话框，该对话框中，选择新建“Windows 应用程序”选项，然后单击“确定”按钮，新建一个 Windows 应用程序。



图 M.2　“新建项目”对话框

（3）选中 Windows 应用程序的默认窗体 Form1，在“属性窗口”中，将其 IsMdiContainer 属性设置为 True，以将该窗体指定为子窗口的多文档界面容器，如图 M.3 所示。

更多资源请访问稀酷客(www.ckook.com)

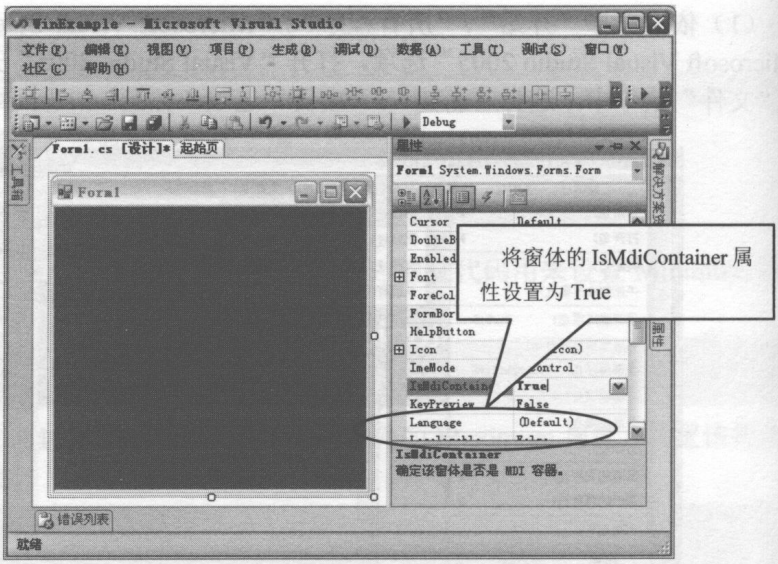


图 M.3　设置窗体的 IsMdiContainer 属性

（4）在 Form1 窗体中添加一个 MenuStrip 控件，以用来从父窗体调用子窗体。这里设置 MenuStrip 控件的一个顶级菜单项和一个子菜单项，如图 M.4 所示。至此，完成 MDI 父窗体的设计。

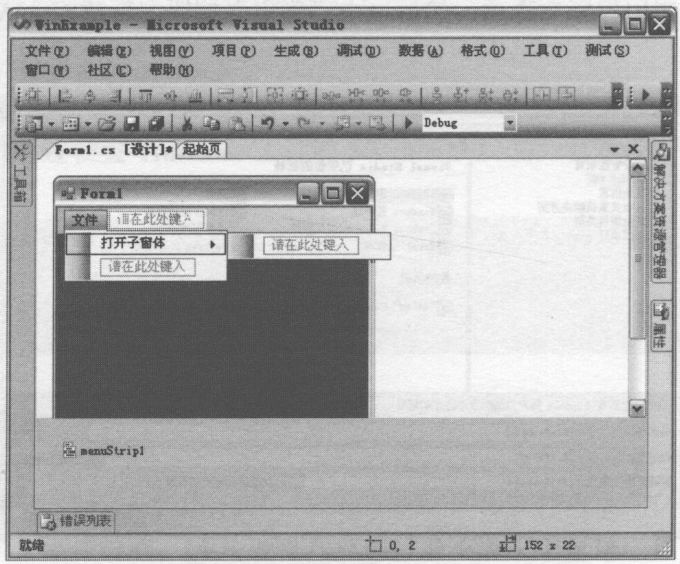


图 M.4　MenuStrip 控件菜单项设置

创建子窗体

MDI 子窗体对于多文档界面应用程序来说是至关重要的，因为用户是通

更多资源请访问稀酷客(www.ckook.com)

过子窗体来与应用程序交互的。创建 MDI 子窗体的方法如下：

在“解决方案资源管理器”中选中含有父窗体的 Windows 应用程序，单击右键，在弹出的快捷菜单中选择“添加”/“新建项”或“添加”/“Windows 窗体”选项，弹出如图 M.5 所示的“添加新项”对话框，该对话框中，选择“Windows 窗体”，并在“名称”文本框中给要添加的 Windows 窗体命名，然后单击“添加”按钮，在当前应用程序中添加一个新的 Windows 窗体。

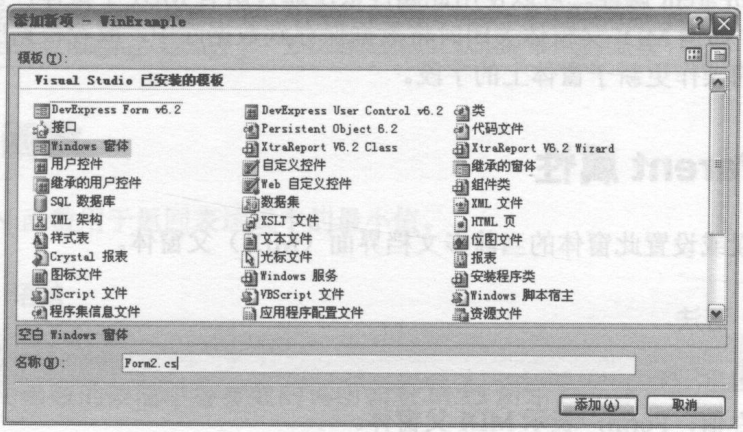


图 M.5 “添加新项”对话框

调用子窗体

这里以前两节中创建的 MDI 父窗体和 MDI 子窗体为例，说明如何从父窗体调用子窗体。选择 MDI 父窗体中 MenuStrip 控件的“打开子窗体”菜单项，触发其 Click 事件，并在该事件下添加如下代码：

```
    private void 打开子窗体ToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        Form2 frmchild = new Form2();
        frmchild.MdiParent = this;
        frmchild.Show();
    }
```

完成上述操作后，运行 Windows 应用程序，当用户单击“打开子窗体”菜单项时，程序就会创建一个新的多文档界面子窗体。

MdiChildren 属性

获取窗体的数组，这些窗体表示以此窗体作为父级的多文档界面 (MDI) 子窗体。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public Form[] MdiChildren { get; }
```

属性值：Form 对象的数组，每个对象都标识此窗体的一个 MDI 子窗体。

此属性可以获取对当前在某 MDI 父窗体中打开的所有 MDI 子窗体的引用。若要创建 MDI 子窗体，请将要成为 MDI 父窗体的 Form 分配给该子窗体的 MdiParent 属性。可以使用此属性依次通过所有 MDI 子窗体，以执行一些操作，如当 MDI 父窗体关闭时将数据保存到数据库中，或者根据应用程序中执行的操作更新子窗体上的字段。

MdiParent 属性

获取或设置此窗体的当前多文档界面（MDI）父窗体。

语法

```
public Form MdiParent { get; set; }
```

属性值：Form，表示 MDI 父窗体。

若要创建 MDI 子窗体，要将成为 MDI 父窗体的 Form 分配给该子窗体的 MdiParent 属性。可以从某 MDI 子窗体使用此属性来获取所有子窗体都需要的全局信息或者调用对所有子窗体执行操作的方法。

Microsoft.Win32 命名空间

Microsoft.Win32 命名空间提供两种类型的类：处理由操作系统引发的事件的类和操作系统注册表的类。

类

Microsoft.Win32 命名空间中包含的类及说明如表 M.12 所示。

表 M.12　Microsoft.Win32 命名空间中包含的类及说明

类	说　明
IntranetZoneCredentialPolicy	定义凭据策略，该凭据策略将用于使用 WebRequest 及其派生类发出的资源请求
PowerModeChangedEventArgs	为 PowerModeChanged 事件提供数据
Registry	提供表示 Windows 注册表中的根项的 RegistryKey 对象，并提供访问项/值对的 static 方法
sRegistryKey	表示 Windows 注册表中的项级节点。此类是注册表封装

更多资源请访问稀酷客(www.ckook.com)

续表

类	说　明
SessionEndedEventArgs	为 SessionEnded 事件提供数据
SessionEndingEventArgs	为 SessionEnding 事件提供数据
SessionSwitchEventArgs	为 SessionSwitch 事件提供数据
SystemEvents	提供对系统事件通知的访问。无法继承此类
TimerElapsedEventArgs	为 TimerElapsed 事件提供数据
UserPreferenceChangedEventArgs	为 UserPreferenceChanged 事件提供数据
UserPreferenceChangingEventArgs	为 UserPreferenceChanging 事件提供数据

MIN 函数

MIN 函数用于返回表达式中的最小值。

语法

```
MIN ( [ ALL ] expression )
```

MIN 函数的语法中各参数的说明如表 M.13 所示。

表 M.13　MIN 函数的语法中各参数的说明

参　数	描　述
ALL	对所有值应用聚合函数，默认值为 ALL
expression	常量、列名或函数以及算术运算符、位运算符和字符串运算符的任意组合

示例

本示例使用 MIN 函数查询学生成绩表（student）中英语成绩（stuEnglish）最低分的学生信息。

```
select *  from student where stuEnglish=(select Min(stuEnglish) from
student)
```

本示例通过 Math 类中的 Min 方法比较指定两个数的大小，返回两个数中较小的数。

```
private void Form1_Load(object sender, EventArgs e)
{
    int aaa = Math.Min(50,100);
}
```

MinDate 属性

获取或设置可在控件中选择的最小日期和时间。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public DateTime MinDate { get; set; }
```

属性值：可在控件中选择的最小日期和时间。默认值为 1/1/1753 00:00:00。

Minimum 属性

获取或设置数字显示框的最小允许值。

语法

```
public decimal Minimum { get; set; }
```

属性值：数字显示框的最小允许值。默认值为 0。

MonthCalendar 控件

MonthCalendar 控件为用户查看和设置日期信息提供了一个直观的图形界面，该控件以网格形式显示日历，网格包含月份的编号日期，这些日期排列在周一到周日下的 7 个列中，并且突出显示选定的日期范围。可以单击月份标题任何一侧的箭头按钮来选择不同的月份，与 DateTimePicker 控件不同的是，开发人员可以使用该控件选择多个日期。

属性

MonthCalendar 控件常用属性及说明如表 M.14 所示。

表 M.14　MonthCalendar 控件常用属性及说明

属　性	说　明
ShowToday	此属性把今天的日期显示在控件底部
ShowTodayCircle	此属性把今天的日期显示为圆形
ShowWeekNumbers	此属性设置为 true，表示在日历中添加周编号
CalendarDimensions	可以水平和垂直显示多个月份
FirstDayOfWeek	根据月历中的显示获取或设置一周中的第一天
SelectionRange	即该控件中选定的日期范围。SelectionRange 值不能超过 MaxSelectionCount 属性中设置的最大可选择天数。用户可以选择的最早和最晚日期由 MaxDate 和 MinDate 属性确定

方法

MonthCalendar 控件常用方法及说明如表 M.15 所示。

更多资源请访问稀酷客(www.ckook.com)

表 M.15　MonthCalendar 控件常用方法及说明

方　法	说　明
AddMonthlyBoldedDate	在月历中，在每月的基础上添加一个以粗体显示的日期
AddBoldedDate	在月历中添加以粗体显示的日期
AddAnnuallyBoldedDate	在月历中，在一年的基础上添加一个以粗体显示的日期
UpdateBoldedDates	重绘粗体格式的日期以反映在粗体格式日期的列表中设置的日期

事件

MonthCalendar 控件常用事件及说明如表 M.16 所示。

表 M.16　MonthCalendar 控件常用事件及说明

事　件	说　明
DateChanged	当 MonthCalendar 中的所选日期更改时发生
DateSelected	用户使用鼠标进行显式日期选择时发生
DoubleClick	当用户双击 MonthCalendar 控件时发生

MouseClick 事件

MouseClick 事件指用户单击控件时发生。

语法

```
public event MouseEventHandler MouseClick
```

Move 方法

Move 方法（Directory 类）

通过 Directory 类的 Move 方法可以对文件夹方便地进行移动。在移动的过程中会将目录及其内容一起移动。

语法：

```
public static void Move (string sourceDirName,string destDirName)
```

Move 方法的语法中各参数的说明如表 M.17 所示。

表 M.17　Move 方法的语法中各参数的说明

参　数	描　述
sourceDirName	要移动的文件或目录的路径
destDirName	指向 sourceDirName 的新位置的路径

更多资源请访问稀酷客(www.ckook.com)

例如，将一个文件夹移动到另一个位置，主要代码如下：

```
Directory.Move("C:\\Soft","D:\\Soft") ;
```

Move 方法（File 类）

实现文件的重命名可以使用 File.Move 方法来实现，即将文件移动到原目录下，只是文件名不同就可以实现更改文件名称。

语法：

```
public static void Move (string sourceFileName,string destFileName)
```

Move 方法的语法中各参数的说明如表 M.18 所示。

表 M.18　Move 方法的语法中各参数的说明

参　数	描　述
sourceFileName	指定欲更名的原文件名
destFileName	为新文件名

注意

因为使用了 File 类，所以要添加对 System.IO 命名空间的引用。

例如，使用 File 类的 Move 方法移动文件，主要代码如下：

```
File.Move("C:\\hb.txt","D:\\hb.txt") ;
```

MoveTo 方法

可以使用 DirectoryInfo 类的 MoveTo 方法移动文件夹。

语法

```
public void MoveTo (string destDirName)
```

MoveTo 方法的语法中各参数的说明如表 M.19 所示。

表 M.19　MoveTo 方法的语法中各参数的说明

参　数	描　述
destDirName	要将此目录移动到的目标位置的名称和路径。目标不能是另一个具有相同名称的磁盘卷或目录，它可以是用户要将此目录作为子目录添加到其中的一个现有目录

更多资源请访问稀酷客(www.ckook.com)

示例

本示例通过 DirectoryInfo 类的 MoveTo 方法将执行的文件夹移动到另一个目录中，如图 M.6 所示。

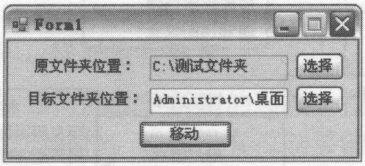


图 M.6　移动文件夹

主要代码如下：

```
    DirectoryInfo DInfo = new DirectoryInfo(textBox1.Text);
    string strPath = textBox2.Text + textBox1.Text.Substring(textBox1.
Text.LastIndexOf("\\") + 1, textBox1.Text.Length - textBox1. Text.
LastIndexOf("\\") - 1);
    DInfo.MoveTo(strPath);
```

本示例通过 MoveTo 方法移动文件到新的位置。

```
    private void Form1_Load(object sender, EventArgs e)
    {
        FileInfo fi = new FileInfo("D:\\200802130212537974.jpg");
        fi.MoveTo("E:\\1.jpg");
    }
```

MVC 开发模式

概念

模型（Model）－视图（View）－控制器（Controller）即为 MVC，MVC 是 Xerox PARC 在 20 世纪 80 年代为编程语言 Smalltalk－80 发明的一种软件架构模式，至今已被广泛使用。

工作原理

MVC 是一个设计模式，它使应用程序的输入、处理和输出强制性分开，使得软件可维护性、可扩展性、灵活性以及封装性得到提高。使用 MVC 应用程序被分成三个核心部件：M（模型）、V（视图）、C（控制器）。模型是所有的商业逻辑代码片段所在。视图表示数据在屏幕上的显示。控制器提供处理过程控制，它在模型和视图之间起连接作用。控制器本身不输出任何信息和做任何处理，它只负责把用户的请求转成针对 Model 的操作，和调用相

更多资源请访问稀酷客(www.ckook.com)

应的视图来显示 Model 处理后的数据。三者之间关系如图 M.7 所示。

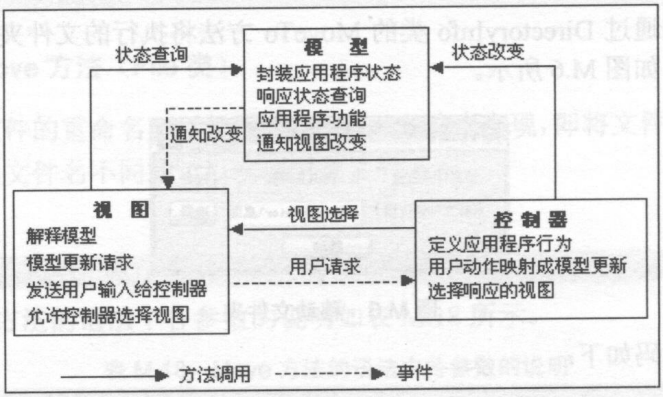


图 M.7 MVC 架构的关系图

MVC（Model-View-Controller）把系统的组成分解为 M（模型）、V（视图）、C（控制器）3 种部件。下面分别对这 3 个核心部件进行介绍。

- 模型

模型表示企业数据和业务规则。在 MVC 的三个部件中，模型拥有最多的处理任务。被模型返回的数据是中立的，就是说模型与数据格式无关，这样一个模型能为多个视图提供数据。由于应用于模型的代码只需写一次就可以被多个视图重用，所以减少了代码的重复性。

- 视图

视图是用户可以看到并与之交互的界面。视图就是由 HTML 元素组成的界面，HTML 依旧在视图中扮演着重要的角色，但一些新的技术已层出不穷，它们包括 Macromedia Flash、XHTML、XML/XSL、WML 等一些标识语言和 Web Services 等。

如何处理应用程序的界面变得越来越有挑战性。MVC 有一个突出的优点是能为应用程序处理很多不同的视图，在视图中其实没有真正的处理发生，不管这些数据是联机存储的还是本地储存的，作为视图来讲，它只是作为一种输出数据并允许用户操纵的方式。

- 控制器

现在总结 MVC 的处理过程，首先控制器接收用户的请求，并决定应该调用哪个模型来进行处理，然后模型用业务逻辑来处理用户的请求并返回数据，最后控制器用相应的视图格式化模型返回的数据，并通过表示层呈现给用户。

更多资源请访问稀酷客(www.ckook.com)



说明

MVC 既有优势，也有劣势，其劣势体现在以下几个方面：

（1）增加了系统结构和实现的复杂性。对于简单的界面，严格遵循 MVC，使模型、视图与控制器分离，会增加结构的复杂性，并可能产生过多的更新操作，降低运行效率。

（2）视图与控制器间的过于紧密的连接。视图与控制器是相互分离，但确实联系紧密的部件，视图没有控制器的存在，其应用是很有限的，反之亦然，这样就妨碍了它们的独立重用。

（3）视图对模型数据的低效率访问。依据模型操作接口的不同，视图可能需要多次调用才能获得足够的显示数据。对未变化数据的不必要的频繁访问，也将损害操作性能。

（4）目前，一般高级的界面工具或构造器不支持 MVC 架构。改造这些工具以适应 MVC 需要和建立分离的部件的代价是很高的，从而造成使用 MVC 的困难。

更多资源请访问稀酷客(www.ckook.com)

N

Navigated 事件

Navigated 事件在 WebBrowser 控件导航到新文档并开始加载该文档时发生。



语法

```
public event WebBrowserNavigatedEventHandler Navigated
```



示例

本示例在 Navigated 事件中，当打开新的网址后，记录输入的 URL。

```
    private void webBrowser1_Na vigated(object
sender,WebBrowserNavigatedEventArgs e)
    {
        toolStripTextBox1.Text = webBrowser1.Url.ToString();
    }
```

Navigate 方法

Navigate 方法用于打开指定的 URL 地址。



语法

```
public void Navigate (string urlString)
```

Navigate 方法的语法中各参数的说明如表 N.1 所示。

表 N.1　Navigate 方法的语法中各参数的说明

参　数	描　述
urlString	要加载的文档的 URL



示例

本示例主要通过调用 WebBrowser 控件的 Navigate 方法、GoForward 方法和 Refresh 方法来操作网页。

```
webBrowser1.Navigate (toolStripTextBox1.Text);//文件导航
webBrowser1.Refresh();//使用控件重新加载页
webBrowser1.GoBack();
webBrowser1.GoForward();
```


更多资源请访问稀酷客(www.ckook.com)

NetworkStream 类

NetworkStream 类提供用于网络访问的基础数据流。

属性

NetworkStream 类的常用属性及说明如表 N.2 所示。

表 N.2 NetworkStream 类的常用属性及说明

属　性	说　明
CanRead	获取一个值，该值指示 NetworkStream 是否支持读取
CanSeek	获取一个值，该值指示流是否支持查找。当前不支持此属性，它始终返回 false
CanTimeout	指示超时属性是否可用于 NetworkStream
CanWrite	获取一个值，该值指示 NetworkStream 是否支持写入
DataAvailable	获取一个值，该值指示在要读取的 NetworkStream 上是否有可用的数据
Length	获取流上可用数据的长度。此属性当前不受支持，总是引发 NotSupportedException
Position	获取或设置流中的当前位置。此属性当前不受支持，总是引发 NotSupportedException
Readable	获取或设置一个值，该值指示 NetworkStream 是否可以读取
ReadTimeout	获取或设置读取操作阻止等待数据的时间量
Socket	获取基础 Socket
Writeable	获取一个值，该值指示 NetworkStream 是否可写
WriteTimeout	获取或设置写入操作阻止等待数据的时间量

方法

NetworkStream 类的常用方法及说明如表 N.3 所示。

表 N.3 NetworkStream 类的常用方法及说明

方　法	说　明
BeginRead	从 NetworkStream 开始异步读取
BeginWrite	开始向流异步写入
Close	关闭 NetworkStream 对象
EndRead	处理异步读取的结束
EndWrite	处理异步写入的结束
Finalize	释放由 NetworkStream 使用的所有资源
Flush	刷新流中的数据。保留此方法供将来使用
Read	从 NetworkStream 读取数据
ReadByte	从流中读取一个字节，并将流内的位置向前推进一个字节，或者如果已到达流的末尾，则返回 –1

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	说　明
Seek	将流的当前位置设置为给定值。此方法当前不受支持，总是引发 NotSupported Exception
SetLength	设置流的长度。此方法始终引发 NotSupportedException
Write	将数据写入 NetworkStream
WriteByte	将一个字节写入流内的当前位置，并将流内的位置向前推进一个字节

new 运算符

功能

new 操作符用于创建一个新的类型实例，它有 3 种形式。

（1）对象创建表达式，用于创建一个类类型或值类型的实例。

（2）数组创建表达式，用于创建一个数组类型实例。

（3）代表创建表达式，用于创建一个新的代表类型实例。

newid 函数

newid 函数用于创建 uniqueidentifier 类型的唯一值。

语法

```
NEWID ( )
```

返回类型：uniqueidentifier。

例如，对变量使用 newid 函数，使用 newid 对声明为 uniqueidentifie 数据类型的变量赋值。在测试该值前，将先打印 uniqueidentifier 数据类型变量的值。

```
DECLARE @myid uniqueidentifier
SET @myid = NEWID()
PRINT 'Value of @myid is: '+ CONVERT(varchar(255), @myid)
```

输出结果为：

```
Value of @myid is: 6F9619FF-8B86-D011-B42D-00C04FC964FF
```

示例

例如：从数据表 tb_Test 中随机抽取 10 条数据，可以利用下面的代码实现。

```
Select top 10 * from tb_Test order by newid()
```


更多资源请访问稀酷客(www.ckook.com)

NewRow 方法

数据绑定控件可以用少量代码实现数据的录入。录入数据时，只需使用 DataTable 类的 NewRow 方法添加一条新记录，然后将信息添加到绑定控件中，录入完成后，使用 Update 方法更新即可完成数据的录入。



语法

```
public DataRow NewRow ()
```



注意

必须使用 NewRow 方法才能创建与 DataTable 具有相同架构的新 DataRow 对象。

Next 方法

Next 方法返回一个指定范围内的随机数。



语法

```
public virtual int Next (int minValue,int maxValue)
```

Next 方法的语法中各参数的说明如表 N.4 所示。

表 N.4　Next 方法的语法中各参数的说明

参　数	描　述
minValue	返回的随机数的下界（随机数可取该下界值）
maxValue	返回的随机数的上界（随机数不能取该上界值）。maxValue 必须大于或等于 minValue
返回值	一个大于或等于 minValue 且小于 maxValue 的 32 位带符号整数，即：返回的值范围包括 minValue 但不包括 maxValue。如果 minValue 等于 maxValue，则返回 minValue

NotifyIcon 控件

NotifyIcon 控件通常用于显示在后台运行的进程的图标，这些进程大部分时间不显示用户界面。每个 NotifyIcon 控件都在状态栏区域显示一个图标。如果有 3 个后台进程，并希望为每个后台进程各显示一个图标，则需要添加 3 个 NotifyIcon 控件。

更多资源请访问稀酷客(www.ckook.com)

属性

NotifyIcon 控件常用属性及说明如表 N.5 所示。

表 N.5 NotifyIcon 控件常用属性及说明

属　性	说　明
Icon	指示图标在任务栏的通知区域中是否可见
Visible	此属性设置确定图控件隐藏还是显示，为使图标显示，Visible 属性必须设置为 true
BalloonTipIcon	此属性获取或设置要显示在与 NotifyIcon 关联的气球提示上的图标
Text	此属性设置当鼠标悬停在该图标上时显示的文字

方法

NotifyIcon 控件常用方法及说明如表 N.6 所示。

表 N.6 NotifyIcon 控件常用方法及说明

方　法	说　明
ShowBalloonTip	在任务栏中显示气球提示

ShowBalloonTip 方法用于在任务栏中持续显示具有指定标题、文本和图标的气球提示指定的时间。

语法：

```
public void ShowBalloonTip (int timeout,string tipTitle,string tipText,ToolTipIcon tipIcon)
```

ShowBalloonTip 方法中的参数及说明如表 N.7 所示。

表 N.7 ShowBalloonTip 方法中的参数及说明

参　数	说　明
timeout	气球提示显示的时间长度（以毫秒为单位）
tipTitle	要在气球提示上显示的标题
tipText	要在气球提示上显示的文本
tipIcon	ToolTipIcon 值之一

事件

NotifyIcon 控件常用事件及说明如表 N.8 所示。

表 N.8 NotifyIcon 控件常用事件及说明

事　件	说　明
DoubleClick	当用户双击任务栏的通知区域中的图标时发生
Click	当用户单击通知区域中的图标时发生
BalloonTipClosed	在用户关闭气球提示时发生

更多资源请访问稀酷客(www.ckook.com)

NumericUpDown 控件

NumericUpDown 控件看起来像是一个文本框与一对箭头的组合，用户可以单击箭头来调整值。NumericUpDown 控件显示并设置选择列表中的单个数值，用户可以通过单击向上和向下按钮、按向上键和向下键或键入一个数字来增大和减小数字。单击向上键时，值沿最大值方向增加；单击向下键时，位置沿最小值方向移动。

属性

NumericUpDown 控件常用属性及说明如表 N.9 所示。

表 N.9　NumericUpDown 控件常用属性及说明

属　性	说　明
DecimalOkaces	指定 NumericUpDown 控件内数据的小数位数
Increment	指定 NumericUpDown 控件递增递减的值
Maximum	指定 NumericUpDown 控件可增加到的最大值
Minimum	指定 NumericUpDown 控件可减小到的最小值
Value	指定 NumericUpDown 控件当前的值

方法

NumericUpDown 控件常用方法及说明如表 N.10 所示。

表 N.10　NumericUpDown 控件常用方法及说明

方　法	说　明
DownButton	将 NumericUpDown 控件的值按照 Increment 属性指定数值递减
UpButton	将 NumericUpDown 控件的值按照 Increment 属性指定数值递增

事件

NumericUpDown 控件常用事件及说明如表 N.11 所示。

表 N.11　NumericUpDown 控件常用事件及说明

事　件	说　明
DoubleClick	在双击控件时发生
Valuechanged	当属性值被修改时，会触发此事件

Net send 命令

在局域网内发送信息，通常使用 Net send 命令进行发送。

更多资源请访问稀酷客(www.ckook.com)

语法

```
Net send {name | * | /DOMAIN[:name] | /USERS} message
```

例如，向局域网中的某个用户（IP 为 192.168.1.27）发送一条信息（“测试数据”）的 net send 命令格式如下。

```
Net send 192.168.1.27 测试数据
```

通过 ProcessStartInfo 类与 Process 组件一起使用执行 DOS 命令（例如“Net send 192.168.1.98 测试数据”）。ProcessStartInfo 类使 Process 类启动进程时，可以访问附加到当前运行进程时可用的进程信息之外的进程信息。

ProcessStartInfo 类常用的属性及说明如表 N.12 所示。

表 N.12 ProcessStartInfo 类的常见属性及说明

属　性	说　明
Arguments	获取或设置启动应用程序时要使用的一组命令行参数
CreateNoWindow	获取或设置指示是否在新窗口中启动该进程的值
Domain	获取或设置一个值，该值标识要在启动进程时使用的域
EnvironmentVariables	获取文件的搜索路径、临时文件的目录、应用程序特定的选项和其他类似信息
ErrorDialog	获取或设置一个值，该值指示不能启动进程时是否向用户显示错误对话框
ErrorDialogParentHandle	获取或设置在为不能启动的进程显示错误对话框时要使用的窗口句柄
FileName	获取或设置要启动的应用程序或文档
LoadUserProfile	获取或设置一个值，该值指示是否从注册表中加载 Windows 用户配置文挡
Password	获取或设置一个安全字符串，该安全字符串包含要在启动进程时使用的用户密码
RedirectStandardError	获取或设置一个值，该值指示是否将应用程序的错误输出写入 Process.StandardError 流中
RedirectStandardInput	获取或设置一个值，该值指示应用程序的输入是否从 Process.StandardInput 流中读取
RedirectStandardOutput	获取或设置一个值，该值指示是否将应用程序的输出写入 Process.StandardOutput 流中
StandardErrorEncoding	获取或设置错误输出的首选编码
StandardOutputEncoding	获取或设置标准输出的首选编码
UserName	获取或设置在启动进程时要使用的用户名
UseShellExecute	获取或设置一个值，该值指示是否使用操作系统外壳程序启动进程
Verb	获取或设置打开 FileName 属性指定的应用程序或文档时要使用的谓词
Verbs	获取与 FileName 属性指定的文件类型关联的一组谓词
WindowStyle	获取或设置启动进程时使用的窗口状态
WorkingDirectory	获取或设置要启动的进程的初始目录

更多资源请访问稀酷客(www.ckook.com)

Net use 命令

Net use 命令关于实现映射网络驱动器。

语法

```
    net use <device name> or * \\<computer name>\<share name>[password
or *]
    [/user:[domain\user] /delete or [persistent:[yes or no]]
    net use <device name> /home /delete or /persistent:[yes or no]
```

Net use 命令的语法中各参数的说明如表 N.13 所示。

表 N.13　Net use 命令的语法中各参数的说明

参　数	描　述
<device name>	要映射的设备名用*代表下一个可用的设备名
\\computer name	控制资源的计算机名
\sharename	共享名
\volume	NetWare server 上的卷名
password	映射的密码*输入密码的提示
/user	<domain>\<user>指定用户以何种身份登录
/home	连接进入用户主目录
/delete	关闭连线
/persistent:[yes or no]	设置用户下一次登录是时是否重新连入

然后通过 ProcessStartInfo 类与 Process 组件一起使用执行 DOS 命令。ProcessStartInfo 类使 Process 类启动进程时，可以访问附加到当前运行进程时可用的进程信息之外的进程信息。

示例

本示例通过 Net use 命令实现映射网络驱动器，运行程序，输入驱动器的名称、网络资源路径，单击“连接”按钮，即可实现映射网络驱动器。运行结果如图 N.1 和 N.2 所示。

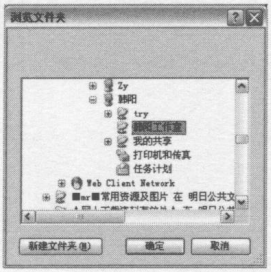


图 N.1　列出工作组中所有计算机 1

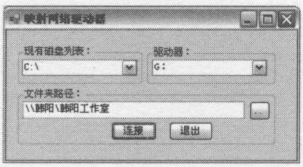


图 N.2　列出工作组中所有计算机 2

更多资源请访问稀酷客(www.ckook.com)

主要代码如下：

```
    private bool ConDir(string Name, string Path)
    {
        try
        {
            System.Diagnostics.ProcessStartInfo psi = new System.
Diagnostics.ProcessStartInfo();
            psi.FileName = @"cmd.exe";
            psi.Arguments = @"/c net use " + Name + " " + Path + "";
            psi.WindowStyle = System.Diagnostics.ProcessWindowStyle.
Hidden;
            System.Diagnostics.Process.Start(psi);
            return true;
        }
        catch (Exception ey)
        {
            return false;
        }
    }
```


更多资源请访问稀酷客(www.ckook.com)

O

object 类型

功能

object 类型在.NET Framework 中是 Object 的别名。在 C#的统一类型系统中，所有类型（预定义类型、用户定义类型、引用类型和值类型）都是直接或间接从 Object 继承的。可以将任何类型的值赋给 object 类型的变量。

示例

本示例来说明 object 类型的引用，从输入结果中可以看到，classRef 只是指向了 a，因为 classRef 并没有实例化，但 classRef.i 却等于 60，如图 O.1 所示。

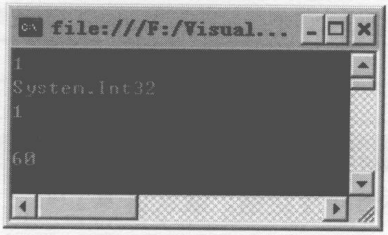


图 O.1　object 类型 a 的引用调用

主要代码如下：

```
using System;
using System.Collections.Generic;
using System.Text;
namespace Ex02_01
{
    class ObjectClass
    {
        public int i = 60;
    }
    class Program
    {
        static void Main(string[] args)
        {
            object a;
```


更多资源请访问稀酷客(www.ckook.com)

```
            a = 1;//为变量 a 赋值
            Console.WriteLine(a);
            Console.WriteLine(a.GetType());
            Console.WriteLine(a.ToString());
            Console.WriteLine();
            a = new ObjectClass();//重新实例化变量 a
            ObjectClass classRef;
            classRef = (ObjectClass)a;//引用调用
            Console.WriteLine(classRef.i);
            Console.Read();
        }
    }
}
```

Opacity 属性

用于获取或设置窗体的不透明度级别。在其他开发环境中，实现窗体的半透明渐显需要调用 API 函数，实现非常困难，但在 C# 2.0 中，窗体提供了 Opacit 属性来设置窗体的透明度。

语法

```
public double Opacity { get; set; }
```

属性值：窗体的不透明度级别。默认值为 1.00。

示例

本示例通过设置 Form 类的 Opacity 属性来实现半透明渐显窗体，如图 O.2 所示。

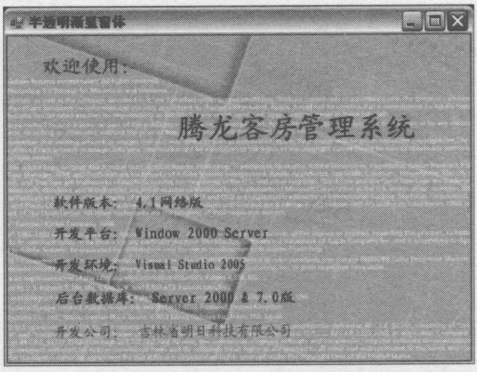


图 O.2　半透明渐显窗体

更多资源请访问稀酷客(www.ckook.com)

主要代码如下：

```
private void timer1_Tick(object sender, EventArgs e)
{
    this.Opacity += 0.1;
}
```

Open 方法

每个函数都可以强制将一个表达式转换成货币型。

Open 方法（File 类）

此方法用于打开指定路径上的 FileStream，具有读/写访问权限。

语法：

```
public static FileStream Open (string path,FileMode mode)
```

Open 方法的语法中各参数的说明如表 O.1 所示。

表 O.1　Open 方法的语法中各参数的说明

参　数	描　述
path	要打开的文件
mode	FileMode 值，用于指定在文件不存在时是否创建该文件，并确定是保留还是改写现有文件的内容
返回值	以指定模式打开的指定路径上的 FileStream，具有读/写访问权限并且不共享

例如，本示例创建一个临时文件并在该文件中写入一些文本。然后使用 Open 方法打开文件，如果文件尚不存在，不会创建该文件，代码如下：

```
public static void Main()
{
    string path = Path.GetTempFileName();
    using (FileStream fs = File.Open(path, FileMode.Open, FileAccess.
Write, FileShare.None))
    {
        Byte[] info = new UTF8Encoding(true).GetBytes("This is some
text in the file.");
        fs.Write(info, 0, info.Length);
    }
    using (FileStream fs = File.Open(path, FileMode.Open))
    {
        byte[] b = new byte[1024];
        UTF8Encoding temp = new UTF8Encoding(true);
        while (fs.Read(b,0,b.Length) > 0)
```


更多资源请访问稀酷客(www.ckook.com)

```
            {
                Console.WriteLine(temp.GetString(b));
            }
        }
    }
```

Open 方法（数据库）

此方法使用 ConnectionString 所指定的属性设置打开数据库连接。

语法：

```
public override void Open ()
```

例如，建立数据的连接，主要通过 SqlConnection 类实现，并初始化数据库连接字符串，然后通过 State 属性判断连接状态，如果数据库连接状态为关，则打开数据库连接。实现打开数据库连接的 Open 方法的代码如下：

```
    private void Open()
    {
        //打开数据库连接
        if (con == null)
        {
            con = new SqlConnection("Data Source=(local);DataBase=db_
CMS;User ID=sa;PWD=");
        }
      if (con.State == System.Data.ConnectionState.Closed)
      con.Open();
    }
```

OpenFile 方法

OpenFile 方法用于打开用户选定的具有只读权限的文件，该文件由 FileName 属性指定。

语法

```
public Stream OpenFile ()
```

返回值：指定用户所选定的只读文件的 Stream。

OpenFileDialog 控件

OpenFileDialog 控件主要用于打开文件，为选择文件提供选择窗口。

更多资源请访问稀酷客(www.ckook.com)

属性

OpenFileDialog 控件常用的属性及说明如表 O.2 所示。

表 O.2 OpenFileDialog 控件常用属性及说明

属　性	说　明
AddExtension	指示如果用户省略扩展名，对话框是否自动在文件名中添加扩展名
CheckFileExists	值指示如果用户指定不存在的文件名，对话框是否显示警告
CheckPathExists	指示如果用户指定不存在的路径，对话框是否显示警告
DefaultExt	获取或设置默认文件扩展名
DereferenceLinks	指示对话框是否返回快捷方式引用的文件的位置，或者是否返回快捷方式（.lnk）的位置
FileName	获取或设置一个包含在文件对话框中选定的文件名的字符串
FileNames	获取对话框中所有选定文件的文件名
Filter	获取或设置当前文件名筛选器字符串，该字符串决定对话框的“另存为文件类型”或“文件类型”框中出现的选择内容
FilterIndex	获取或设置文件对话框中当前选定筛选器的索引
InitialDirectory	获取或设置文件对话框显示的初始目录
Multiselect	获取或设置一个值，该值指示对话框是否允许选择多个文件
ReadOnlyChecked	获取或设置一个值，该值指示是否选定只读复选框
RestoreDirectory	获取或设置一个值，该值指示对话框在关闭前是否还原当前目录
ShowHelp	获取或设置一个值，该值指示文件对话框中是否显示“帮助”按钮
ShowReadOnly	获取或设置一个值，该值指示对话框是否包含只读复选框
Title	获取或设置文件对话框标题

方法

OpenFileDialog 控件常用的方法及说明如表 O.3 所示。

表 O.3 OpenFileDialog 控件常用方法及说明

方　法	说　明
OpenFile	此方法以只读模式打开用户选择的文件
ShowDialog	此方法显示 OpenFileDialog

事件

OpenFileDialog 控件常用的事件及说明如表 O.4 所示。

表 O.4 OpenFileDialog 控件常用事件及说明

事　件	说　明
FileOk	当用户单击对话框中“打开”或“保存”按钮时触发的事件
HelpRequest	当用户单击对话框中的“帮助”按钮时将触发的事件

更多资源请访问稀酷客(www.ckook.com)

OpenSubKey 方法

Registry 类的 OpenSubKey 方法用于检索指定的子项。

语法

```
public RegistryKey OpenSubKey(string name)
```

OpenSubKey 方法的语法中各参数的说明如表 O.5 所示。

表 O.5　OpenSubKey 方法的语法中各参数的说明

参　数	描　述
name	要以只读方式打开的子项的名称或路径
返回值	请求的子项；如果操作失败，则为空引用

OpenText 方法

OpenText 方法用于打开现有 UTF-8 编码文本文件以进行读取。

语法

```
public static StreamReader OpenText (string path)
```

OpenText 方法的语法中各参数的说明如表 O.6 所示。

表 O.6　OpenText 方法的语法中各参数的说明

参　数	描　述
path	要打开以进行读取的文件
返回值	指定路径上的 StreamReader

示例

本示例通过 OpenText 方法打开现有 UTF-8 编码文本文件并读取，如图 O.3 所示。

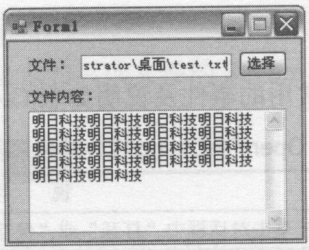


图 O.3　打开现有 UTF-8 编码文本文件并进行读取

更多资源请访问稀酷客(www.ckook.com)

主要代码如下：

```
openFileDialog1.Filter = "文本文件(*.txt)|*.txt";
openFileDialog1.ShowDialog();
textBox1.Text = openFileDialog1.FileName;
StreamReader SReader = File.OpenText(textBox1.Text);
textBox2.Text = SReader.ReadToEnd();
```

Owner 属性

Form.Owner 属性用来获取或设置拥有此窗体的窗体。

语法

```
public Form Owner {get; set;}
```

属性值：Form，表示作为此窗体的所有者的窗体。

说明

若将某窗体设置为另一个窗体所有时，请为其 Owner 属性分配一个对将成为所有窗体的引用。当一个窗体归另一窗体所有时，它便随着所有者窗体最小化或关闭。例如，如果窗体 Form2 归窗体 Form1 所有，则关闭或最小化 Form1 时，Form2 也关闭或最小化，并且 Form2 窗体不显示在任务栏中。

示例

本示例通过 Owner 属性设置窗体的所有者，代码如下：

```
//调用 AuthUpdate 方法修改明星信息
if (tbAuMenthod.AuthUpdate(tbAu) == 1)
{
    MessageBox.Show("修改成功");
    frmMain frm = (frmMain)this.Owner;
    tbAuMenthod.tb_authorinfoFill("2", frm.lvAuton);
    intFalg = 0;
    this.Close();
}
```


更多资源请访问稀酷客(www.ckook.com)

P

Paint 事件

Paint 事件在重绘或更新控件时发生。

语法

```
public event PaintEventHandler Paint
```

Panel 控件

Panel 控件用于为其他控件提供可识别的分组。开发 Windows 应用程序时，通常使用 Panel 控件按功能细分窗体。在窗体设计时，所有控件都可以自由移动，而当移动 Panel 控件时，它包含的所有控件也将随着移动。

属性

Panel 控件的常用属性及说明如表 P.1 所示。

表 P.1　Panel 控件常用属性及说明

属　性	说　明
AutoScroll	此属性指示当控件内容大于它的可见区域时是否自动显示滚动条
BackColor	此属性获取或设置控件的背景色
BackgroundImage	此属性获取或设置在控件中显示的背景图像
BorderStyle	此属性指示控件的边框样式

方法

Panel 控件常用方法及说明如表 P.2 所示。

表 P.2　Panel 控件常用方法及说明

方　法	说　明
Show	该方法主要用于向用户显示控件
Focus	为控件设置输入焦点

下面对 Show 方法进行详细介绍，该方法主要用于向用户显示控件。

语法：

```
public void Show ()
```


更多资源请访问稀酷客(www.ckook.com)



注意

显示控件等效于将 Visible 属性设置为 true。在调用 Show 方法之后，只要不调用 Hide 方法，Visible 属性就返回 true 值。

PasswordChar 属性

PasswordChar 属性用于获取或设置要取代用户输入而显示的字符。



语法

```
public char PasswordChar { get; set; }
```

属性值：用做密码字符的 Char 值。



示例

本示例通过设置 TextBox 控件的 PasswordChar 属性，使用户输入的密码显示为“*”。示例运行效果如图 P.1 所示。

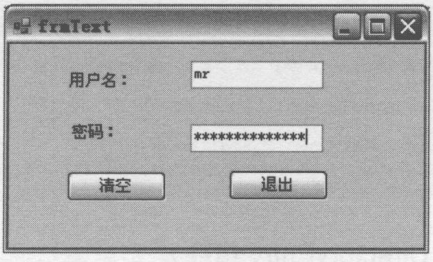


图 P.1　PasswordChar 属性

主要代码如下：

```
private void Form2_Load(object sender, EventArgs e)
{
    textBox1.PasswordChar = '*';
    textBox1.MaxLength = 14;
}
```

除了 TextBox 控件有 PasswordChar 属性外，当向窗体中添加一个 ToolStrip 控件后，在该控件中添加一个 ToolStripTextBox，如果此文本框用于密码输入，那么就要设置它的 PasswordChar 属性，它的 PasswordChar 属性与 TextBox 控件稍微有些不同，代码如下：

更多资源请访问稀酷客(www.ckook.com)

```
private void Form1_Load(object sender, EventArgs e)
{
    toolStripTextBox1.TextBox.PasswordChar = "*";
}
```

Path 属性

Path 属性（FileSystemWatcher 组件）

Path 属性获取或设置要监视的目录的路径。

语法：

```
public string Path { get; set; }
```

属性值：要监视的路径。默认值为空字符串（""）。

例如，通过 FileSystemWatcher 组件，用来监视系统日志文件，设置其 path 属性用于指定监视的目录。

```
private void button1_Click(object sender, EventArgs e)
{
    fileSystemWatcher1.Path = textBox1.Text;
    //此属获取或设置筛选字符串，用于确定在目录中监视哪些文件
    this.fileSystemWatcher1.Filter = "*.Evt";
    this.fileSystemWatcher1.EndInit();
}
```

Path 属性（DirectoryEntry 组件）

DirectoryEntry 类封装 Active Directory 层次结构中的节点或对象，使用该类可以绑定到对象，或者读取和更新属性。

Path 属性用于获取或设置 DirectoryEntry 对象的路径，默认值为空字符串（""）。

语法：

```
public string Path { get; set; }
```

属性值：DirectoryEntry 对象的路径，默认值为空字符串（""）。

例如，设置 Path 属性，将本机上的用户名，工作组添加到 treeview 控件中。其运行结果如图 P.2 所示。

更多资源请访问稀酷客(www.ckook.com)

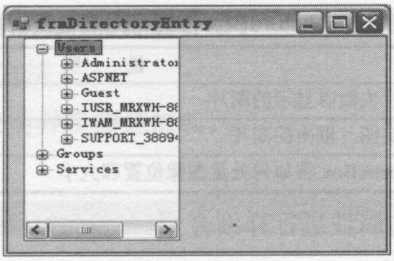


图 P.2　Path 属性

主要代码如下：

```
private void frmDirectoryEntry_Load(object sender, EventArgs e)
{
    entryPC.Path = "WinNT://workgroup/Localhost";
    TreeNode users = new TreeNode("Users");
    TreeNode groups = new TreeNode("Groups");
    TreeNode services = new TreeNode("Services");x
    viewPC.Nodes.AddRange(new TreeNode[] { users, groups, services });
    foreach (DirectoryEntry child in entryPC.Children)
    {
        TreeNode newNode = new TreeNode(child.Name);
        switch (child.SchemaClassName)
        {
            case "User":
                users.Nodes.Add(newNode);
                break;
            case "Group":
                groups.Nodes.Add(newNode);
                break;
            case "Service":
                services.Nodes.Add(newNode);
                break;
        }
        AddPathAndProperties(newNode, child);
    }
}
```

PictrueBox 控件

PictureBox 控件用于显示位图、GIF、JPEG、图元文件和图标格式的图形。

属性

PictureBox 控件常用的属性及说明如表 P.3 所示。

更多资源请访问稀酷客(www.ckook.com)

表 P.3　PictureBox 控件常用属性及说明

属　性	说　明
ErrorImage	加载图片失败时显示的图片
Image	用于指定图片框中的图片
SizeMode	控件 PictureBox 将如何处量图像位置和大小

下面对比较重要的属性进行详细介绍。

（1）SizeMode 属性

此属性用于设置 PictureBox 控件将如何处理图像位置和大小。

语法：

```
public PictureBoxSizeModeSizeMode { get; set; }
```

属性值：PictureBoxSizeMode 值之一，默认为 Normal。

PictureBoxSizeMode 是一个返回值，用来处理图像位置和大小。PictureBoxSizeMode 的属性值及说明如表 P.4 所示。

表 P.4　PictureBoxSizeMode 的属性值及说明

属 性 值	说　明
AutoSize	调整 PictureBox 大小，使其等于所包含的图像大小则图像将居中显示。如果图像比 PictureBox 大，则图片将居于 PictureBox 中心，而外边缘将被剪裁
CenterImag	如果 PictureBox 比图像大，则图像将居中显示
Normal	图像被置于 PictureBox 的左上角。如果图像比包含它的 PictureBox 大，则该图像将被剪裁掉
StretchImag	PictureBox 中的图像被拉伸或收缩，以适合 PictureBox 的大小
Zoom	图像大小按其原有的大小比例被增加或减小

（2）Image 属性

此属性用于指定图片框中的图片。

语法：

```
public Image Image { get; set; }
```

属性值：要显示的 Image。

方法

PictureBox 控件常用方法及说明如表 P.5 所示。

表 P.5　PictureBox 控件常用方法及说明

方　法	说　明
Show	向用户显示控件
Load	在 PictureBox 中显示图像
CancelAsyn	该方法主要用于取消异步图像加载

更多资源请访问稀酷客(www.ckook.com)

下面对比较重要的方法进行详细介绍。

（1）CancelAsync 方法

该方法主要用于取消异步图像加载。

语法：

```
public void CancelAsync ()
```

（2）Load 方法

在 PictureBox 中显示图像。

语法：

```
public void Load (string url)
```

Load 方法的语法中各参数的说明如表 P.6 所示。

表 P.6　Load 方法的语法中各参数的说明

参　数	描　述
url	要在 PictureBox 中显示的图像的路径



事件

PictureBox 控件常用事件及说明如表 P.7 所示。

表 P.7　PictureBox 控件常用事件及说明

事　件	说　明
SizeModeChanged	当 SizeMode 属性修改时，会发生此事件。SizeMode 属性可通过编程方式修改
TextChanged	当修改 Text 属性被修改时，发生的事件
Click	当单击图片时发生的事件

下面详细介绍 Click 事件，该事件在单击图片时发生。

语法：

```
public event EventHandler Click
```

例如，当程序运行的时候，单击时钟图片，在窗体下方显示当前时间。实例运行结果如图 P.3 所示。



图 P.3　Click 事件的使用

更多资源请访问稀酷客(www.ckook.com)

程序主要代码如下：

```
private void pictureBox1_Click(object sender, EventArgs e)
{
    timer1.Interval = 1000;                    //设置时间间隔
    timer1.Enabled = true;                     //启用计时器
}
private void timer1_Tick(object sender, EventArgs e)
{
    label1.Text = "现在时间：" + DateTime.Now.ToString();
}
```

POP3 协议

常用命令

POP（Post Office Protocol 邮局协议）用于电子邮件的接受，现在常用第3版，所以称 POP3。通过 POP 协议，客户机登录到服务器后，可以对自己的邮件进行删除，或是下载到本地。表 P.8 为 POP3 的常用命令。

表 P.8　POP3 常用命令

命　令	描　述
USER	此命令与下面的 pass 命令若都发送成功，将使状态转换
PASS	用户名所对应的密码
APOP	Digest 是 MD5 消息摘要
STAT	请求服务器发回关于邮箱统计资料（邮件总数和总字节数）
UIDL	回送邮件唯一标识符
LIST	回送邮件数量和每个邮件的大小
RETR	回送由参数标识的邮件的全部文本
DELE	服务器将由参数标识的邮件标记为删除，由 QUIT 命令执行
RSET	服务器将重置所有标记为删除的邮件，用于撤销 DELE 命令
TOP	服务器将回送由参数标识的邮件前 n 行内容，n 是正整数
NOOP	服务器返回一个肯定的响应，不做任何操作
QUIT	退出

注意

SMTP 服务器使用的端口号一般为 25，POP 服务器使用的端口号一般为 110。

更多资源请访问稀酷客(www.ckook.com)

配置 POP3

实现电子邮件的接收功能时，首先需要配置 POP3 服务，步骤如下：

（1）首先在“Windosw 组件向导”对话框中选中“电子邮件服务”前的复选框，依次单击“下一步”按钮，完成“电子邮件服务”的添加操作。“Windosw 组件向导”对话框如图 P.4 所示。

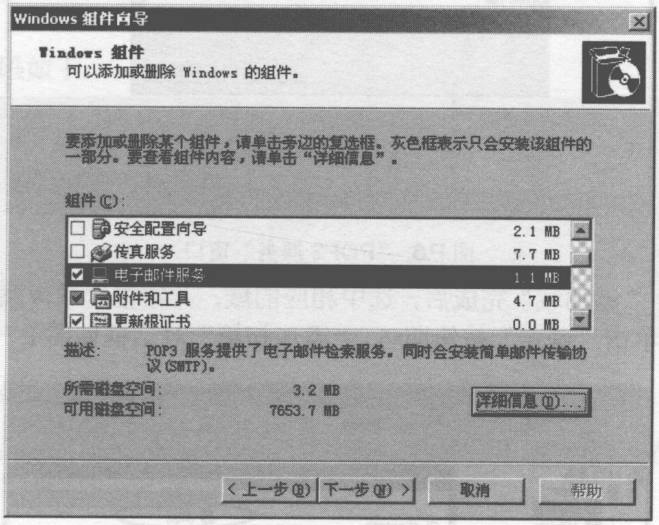


图 P.4 “Windosw 组件向导”对话框

（2）当添加完“电子邮件服务”后，打开“管理工具”，这时将会出现一项新的功能“POP3 服务”，如图 P.5 所示。

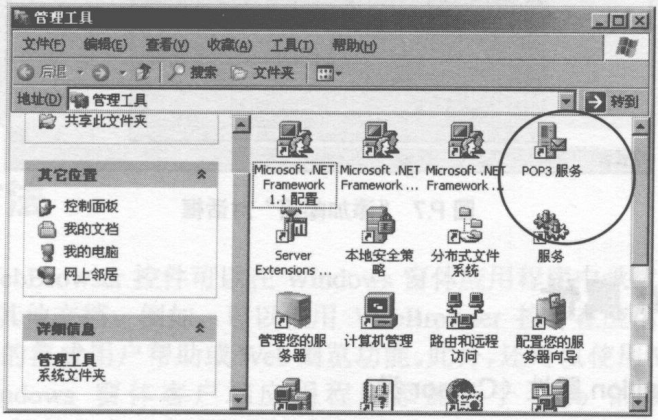


图 P.5 “管理工具”窗体

（3）双击打开“POP3 服务”，添加一个新域（如“126.com”），单击“确定”按钮，如图 P.6 所示。

更多资源请访问稀酷客(www.ckook.com)

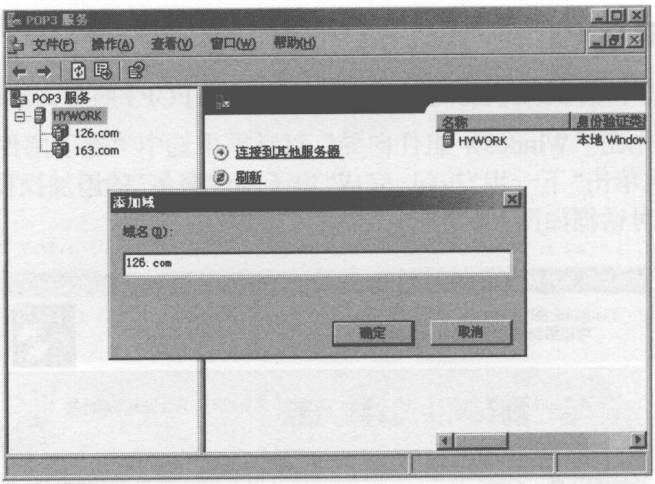


图 P.6 “POP3 服务”窗口

（4）当“添加域”完成后，选中相应的域，并在指定域内添加邮箱名和密码，然后单击“确定”按钮即可。“添加邮箱”对话框如图 P.7 所示。

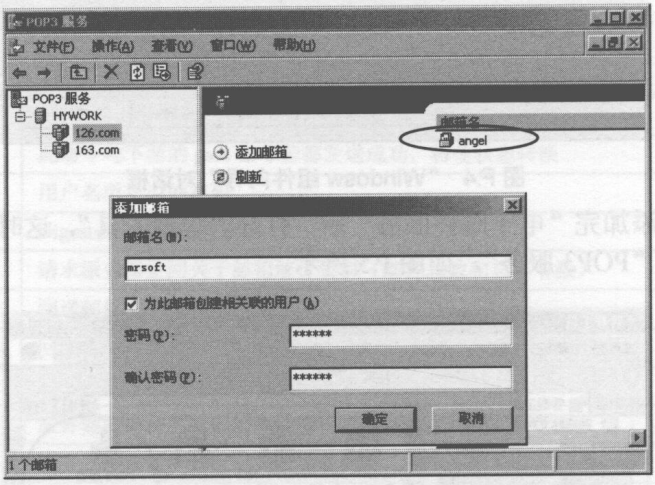


图 P.7 “添加邮箱”对话框

Position 属性

Position 属性（Cursor 类）

Cursor 类的 Position 属性用于获取或设置光标位置。

语法：

```
public static Point Position { get; set; }
```


更多资源请访问稀酷客(www.ckook.com)

属性值：代表光标位置的 Point（采用屏幕坐标）。

例如，将鼠标移动到窗体外，同时按下 Alt+W 键，效果如图 P.8 所示。

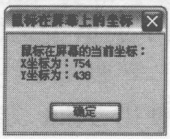


图 P.8　屏幕上鼠标的坐标

主要代码如下：

```
private void button1_Click(object sender, EventArgs e)
{
    string str = "鼠标在屏幕的当前坐标：\n";
    str += "X 坐标为：" + Cursor.Position.X.ToString() +"\n";
    str += "Y 坐标为：" + Cursor.Position.Y.ToString()+ "\n";
    MessageBox.Show(str, "鼠标在屏幕上的坐标");
}
```

Position 属性（FileStream 类）

FileStream 类的 Position 属性用于获取或设置此流的当前位置。

语法：

```
public override long Position { get; set; }
```

属性值：此流的当前位置。

本示例使用 Length 和 Position 属性检查文件尾条件。

```
if( s.Length==s.Position )
{
   Console.WriteLine("End of file has been reached.");
}
```

Print 方法

使用 WebBrowser 控件可以在 Windows 窗体应用程序中承载网页以及支持浏览器的其他文档。例如，可以使用 WebBrowser 控件在应用程序中提供基于 HTML 的集成用户帮助或 Web 浏览功能。此外，还可以使用 WebBrowser 控件向 Windows 窗体客户端应用程序添加基于 Web 的现有控件。WebBrowser 控件包含多个可以用来实现 Internet Explorer 中的控件的属性、方法和事件。Print 方法使用当前打印和页面设置打印当前显示在 WebBrowser 控件中的文档。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public void Print ()
```

示例

下面的代码示例演示如何使用 Print 方法为 WebBrowser 控件实现与 Internet Explorer 中的“打印”按钮类似的“打印”按钮，如图 P.9 所示。

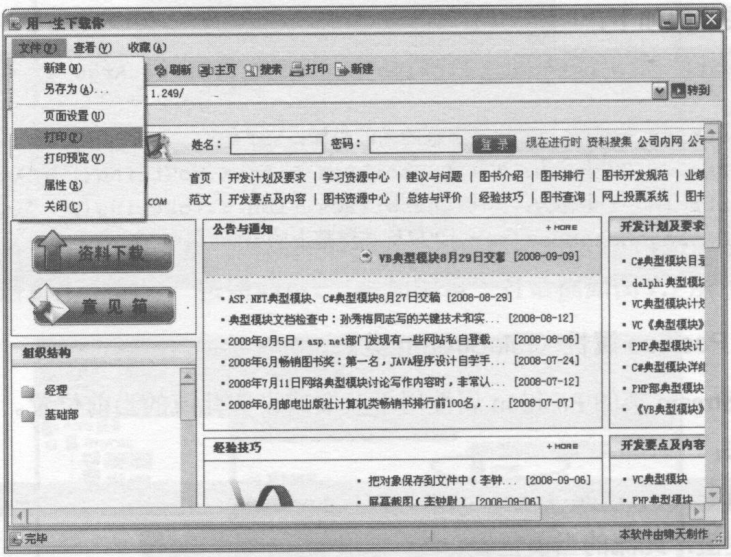


图 P.9 通过 Print 方法打印当前页面

主要代码如下：

```
private void printButton_Click(object sender, EventArgs e)
{
    webBrowser1.Print();                          //打印当前页
}
```

PrintDialog 组件

Windows 窗体 PrintDialog 组件是一个预先配置的对话框，可用于在基于 Windows 应用程序中选择打印机、选择要打印的页以及确定其他与打印相关的设置。将该控件用做选择打印机和打印相关设置的简单解决方案，而不用配置您自己的对话框。可使用户能够打印文档的很多部分：全部打印、打印指定的页范围或打印选定内容。利用标准的 Windows 对话框，可以创建其基本功能和立即为用户所熟悉的应用程序。

更多资源请访问稀酷客(www.ckook.com)

从应用程序的代码内调用 ShowDialog 方法，可以显示打印对话框。在显示该组件后，用户将与之进行交互，设置打印作业的属性。这些项保存在与该打印作业关联的 PrinterSettings 类和 PageSettings 类（如果用户通过 PrintDialog 组件访问 PageSetupDialog 组件（Windows 窗体））中后，可以调用它们所设置的属性以确定打印作业的细节。

属性

PrintDialog 组件的属性及说明如表 P.9 所示。

表 P.9 PrintDialog 组件的属性及说明

属　性	说　明
AllowCurrentPage	获取或设置一个值，该值指示是否显示“当前页”选项按钮
AllowPrintToFile	获取或设置一个值，该值指示是否启用“打印到文件”复选框
AllowSelection	获取或设置一个值，该值指示是否启用“选择”选项按钮
AllowSomePages	获取或设置一个值，该值指示是否启用“页”选项按钮
Document	获取或设置一个值，指示用于获取 PrinterSettings 类的 PrintDocument 对象
PrinterSettings	获取或设置对话框修改的打印机设置
PrintToFile	获取或设置一个值，该值指示允许用户选择的最小边距（以百分之一英寸为单位）。获取或设置一个值，该值指示是否选中“打印到文件”复选框
ShowHelp	获取或设置一个值，该值指示是否显示“帮助”按钮
ShowNetwork	获取或设置一个值，该值指示是否显示“网络”按钮

下面对比较重要的属性进行详细介绍。

（1）Document 属性

用于获取或设置一个值，指示用于获取 PrinterSettings 类的 PrintDocument 对象。

语法：

```
public PrintDocument Document { get; set; }
```

属性值：用于获取 PrinterSettings 的 PrintDocument，默认为空引用。

（2）AllowCurrentPage 属性

获取或设置一个值，该值指示是否显示“当前页”选项按钮。

语法：

```
public bool AllowCurrentPage { get; set; }
```

属性值：如果显示“当前页”选项按钮，为 true；否则为 false。默认为 false。

（3）AllowPrintToFile 属性

获取或设置一个值，该值指示是否启用“打印到文件”复选框。

更多资源请访问稀酷客(www.ckook.com)

语法：

```
public bool AllowPrintToFile { get; set; }
```

属性值：如果启用“打印到文件”复选框，为 true；否则为 false。默认为 true。

（4）AllowSelection 属性

获取或设置一个值，该值指示是否启用“选择”选项按钮。

语法：

```
public bool AllowSelection { get; set; }
```

属性值：如果启用“选择”选项按钮，为 true；否则为 false。默认为 false。

方法

PrintDialog 组件的方法及说明如表 P.10 所示。

表 P.10　PrintDialog 组件的方法及说明

方　法	说　明
Reset	将所有选项、最后选定的打印机和页面设置重新设置为其默认值
ShowDialog	已重载。运行通用对话框

下面对 ShowDialog 方法进行详细介绍。

PrintDialog 组件中最常用的方法是 ShowDialog 方法，用来显示与打印机相关的设置的对话框，常与 Document 属性一起使用。

语法：

```
public DIALOGRESULT ShowDialog ()
```

如果用户在对话框中单击“确定”，则为 DialogResult.OK；否则为 DialogResult.Cancel。

PrintDocument 组件

Windows 窗体 PrintDocument 组件用于设置一些属性，这些属性说明在基于 Windows 的应用程序中要打印什么内容以及打印文档的能力，可将它与 PrintDialog 组件一起使用来控制文档打印的各个方面。

PrintDocument 是最重要的 Windows 打印对象，它负责建立起和其他打印对象的联系。要想建立输出到打印机的输出程序，就要首先创建 PrintDocument 对象。

属性

PrintDocument 组件的常用属性及说明如表 P.11 所示。

更多资源请访问稀酷客(www.ckook.com)

表 P.11　PrintDocument 组件的常用属性及说明

属　性	说　明
DefaultPageSettings	获取或设置页设置，这些页设置用做要打印的所有页的默认设置
DocumentName	获取或设置打印文档时要显示的文档名（例如，在打印状态对话框或打印机队列中显示）
OriginAtMargins	获取或设置一个值，该值指示与页关联的图形对象的位置是位于用户指定边距内，还是位于该页可打印区域的左上角
PrintController	获取或设置指导打印进程的打印控制器
PrinterSettings	获取或设置对文档进行打印的打印机



方法

PrintDocument 组件的常用方法及说明如表 P.12 所示。

表 P.12　PrintDocument 组件的常用方法及说明

方　法	说　明
Print	开始文档的打印进程

下面对比较重要的方法 Print()方法进行详细介绍。

Print 方法是最重要也是最常用的方法，它将开始打印进程，并触发 PrintPage 事件。在 PrintPage 事件绘制将要打印的内容。

语法：

```
public DIALOGRESULT ShowDialog () public void Print ()
```

指定打印输出的方法是处理 PrintPage 事件并使用 PrintPageEventArgs 参数中包含的 Graphics 对象。

使用 PrinterSettings.PrinterName 属性可指定用哪一台打印机来打印文档。Print 方法在打印文档时不使用打印对话框。若要为用户提供选择打印设置的能力，则使用 PrintDialog 组件。



事件

PrintDocument 组件的常用事件及说明如表 P.13 所示。

表 P.13　PrintDocument 组件的常用事件及说明

事　件	说　明
BeginPrint	在调用 Print 方法时并且在打印文档的第一页之前发生
EndPrint	打印完文档的最后一页时发生
PrintPage	当需要为当前页打印的输出时发生
QueryPageSettings	在 PrintPage 事件的紧面前发生

更多资源请访问稀酷客(www.ckook.com)

下面详细介绍 PrintPage 事件。

当需要为当前页打印的输出时发生。

语法：

```
public event PrintPageEventHandler PrintPage
```

这些事件中最常用的是 PrintPage 事件，该事件应当与 PrintPageEvent Handle 参数委托的签名匹配。

```
Public delegate void PrintPageHandler(object
sender,PrintPageEventArgs e);
```

PrintPageEventArgs 参数为打印提供必要的系统数据，相关参数的说明如表 P.14 所示。

表 P.14 PrintPageEventArgs 参数列表及说明

参　数	说　明
Cancel	布尔值，设置为 true 则取消打印
Graphics	用于写到打印机的 Graphics 对象
HasMorePages	布尔值，指示是否还有待打印的页面。默认为 false
MarginBounds	表示整个页面的矩形区域
PageBounds	表示整个页面的矩形区域
PageSettings	待打印页面的设置

下面对 PrintPage 事件的 Graphics 参数进行详细介绍。

PrintPageEventArgs 参数的成员，包含一个 Graphics 对象用来绘制将要打印的文字和图像。

语法：

```
public Graphics Graphics { get; }
```

这些参数中最重要的也是最常用的是 Graphics 属性，它用来绘制写到屏幕或打印机的文字或图形。例如：

```
    private void printDocument1_PrintPage(object sender, System.
Drawing.Printing.PrintPageEventArgs e)
    {
    e.Graphics.FillRectangle(Brushes.Red, new Rectangle(150, 500, 500,
500));
    }
```

PrintPreviewControl 组件

Windows 窗体 PrintPreviewControl 组件用于按文档打印时的外观显示 PrintDocument 组件。此控件没有按钮或其他用户界面元素，表示打印预览的原

更多资源请访问稀酷客(www.ckook.com)

始预览部分，没有任何对话框或按钮。大多数 PrintPreviewControl 对象都可在 PrintPreviewDialog 对象上找到。因此通常只有在希望编写自己的打印预览用户界面时才使用 PrintPreviewControl 组件。一般都直接使用 PrintPreviewDialog 组件。

属性

PrintPreviewControl 组件的常用属性及说明如表 P.15 所示。

表 P.15 PrintPreviewControl 组件的常用属性及说明

属　性	说　明
Document	获取或设置一个值，该值指示要预览的文档
Enabled	获取或设置一个值，该值指示控件是否可以对用户交互作出响应
MaximumSize	获取或设置大小，该大小是 GetPreferredSize 方法可以指定的上限
MinimumSize	获取或设置大小，该大小是 GetPreferredSize 方法可以指定的下限
Rows	获取或设置屏幕纵向显示的页数
StartPag	获取或设置页面左上角的页码
UseAntiAlias	获取或设置一个值，该值指示打印是否使用操作系统的防锯齿功能
Visible	获取或设置一个值，该值指示是否显示该控件
Zoom	获取或设置一个值，该值指示页面的显示大小

方法

PrintPreviewControl 组件的常用方法及说明如表 P.16 所示。

表 P.16 PrintPreviewControl 组件的常用方法及说明

方　法	说　明
CreateGraphics	为控件创建 Graphics 类
Focus	为控件设置输入焦点
Hide	对用户隐藏控件
PointToClient	将指定屏幕点的位置计算成工作区坐标
PointToScreen	将指定工作区点的位置计算成屏幕坐标
ResetBackColor	将 BackColor 属性重置为其默认颜色
Show	向用户显示控件

PrintPreviewDialog 组件

Windows 窗体 PrintPreviewDialog 组件是预先配置的对话框，用于显示 PrintDocument 组件在打印时的外观。可在基于 Windows 的应用程序中使用它作为简单的解决方案，而不用配置自己的对话框。该控件包含打印、放大、显示一页或多页和关闭此对话框的按钮。

更多资源请访问稀酷客(www.ckook.com)

有些属性可通过 PrintPreviewDialog 组件包含的 PrintPreviewControl 对象获得。（不必向窗体添加此 PrintPreviewControl 组件，向窗体添加 PrintPreviewDialog 对话框时它自动包含在此对话框中。）可通过 PrintPreviewControl 组件的 Columns 和 Rows 属性设置在控件上水平和垂直显示的页的数目。PrintPreviewDialog 组件的属性和方法和 PrintPreviewControl 的属性和方法基本相同。

属性

PrintPreviewDialog 组件的常用属性及说明如表 P.17 所示。

表 P.17　PrintPreviewDialog 组件的常用属性及说明

属　性	说　明
ContextMenu	获取或设置控件的快捷菜单
ContextMenuStrip	获取或设置该控件的快捷菜单
ControlBox	获取或设置一个值，该值指示在该窗体的标题栏中是否显示控件框
Document	获取或设置要预览的文档
Enabled	获取或设置一个值，该值指示是否已启用该控件
Font	获取或设置用于控件的字体
ForeColor	获取或设置控件的前景色
MaximizeBox	获取或设置一个值，该值指示是否在窗体的标题栏中显示最大化按钮
MaximumSize	获取或设置窗体可调整到的最大大小
UseAntiAlias	获取或设置一个值，该值指示打印是否使用操作系统的防锯齿功能

方法

PrintPreviewDialog 组件的常用方法及说明如表 P.18 所示。

表 P.18　PrintPreviewDialog 组件的常用方法及说明

方　法	说　明
Hide	获取一个值，该值指示是否可以选中控件
Show	显示窗体
ShowDialog	将窗体显示为模式对话框

Process 类

Process 类提供了对正在计算机上运行的进程的访问。用最简短的话来说，进程就是当前计算机运行的应用程序。线程是操作系统向其分配处理器时间的基本单位。线程可执行进程的任何一部分代码，包括当前由另一线程执行的部分。对于启动、停止、控制和监视应用程序等任务，Process 类是很

更多资源请访问稀酷客(www.ckook.com)

有用的工具。使用它不但可以获取正在计算机中运行的进程列表，而且还可以启动新的进程。

属性

Process 类的常用属性及说明如表 P.19 所示。

表 P.19　Process 类的常用属性及说明

属　性	说　明
BasePriority	获取关联进程的基本优先级
ExitCode	获取关联进程终止时指定的值
ExitTime	获取关联进程退出的时间
Handle	返回关联进程的本机句柄
HandleCount	获取由进程打开的句柄数
HasExited	获取指示关联进程是否已终止的值
Id	获取关联进程的唯一标识符
MachineName	获取关联进程正在其上运行的计算机的名称
MainModule	获取关联进程的主模块
MainWindowHandle	获取关联进程主窗口的窗口句柄
MainWindowTitle	获取进程的主窗口标题
MaxWorkingSet	获取或设置关联进程的允许的最大工作集大小
MinWorkingSet	获取或设置关联进程的允许的最小工作集大小
Modules	获取已由关联进程加载的模块
NonpagedSystemMemorySize	获取分配给此进程的未分页的系统内存大小
NonpagedSystemMemorySize64	获取为关联的进程分配的非分页系统内存量
PagedMemorySize	获取分页的内存大小
PagedMemorySize64	获取为关联的进程分配的分页内存量
PagedSystemMemorySize	获取分页的系统内存大小
PagedSystemMemorySize64	获取为关联的进程分配的可分页系统内存量
PeakPagedMemorySize	获取峰值分页内存大小
PeakPagedMemorySize64	获取关联的进程使用的虚拟内存分页文件中的最大内存量
PeakVirtualMemorySize	获取峰值虚拟内存大小
PeakVirtualMemorySize64	获取关联的进程使用的最大虚拟内存量
PeakWorkingSet	获取关联进程的峰值工作集大小
PeakWorkingSet64	获取关联的进程使用的最大物理内存量
PriorityBoostEnabled	获取或设置一个值，该值指示主窗口拥有焦点时是否由操作系统暂时提升关联进程的优先级
PriorityClass	获取或设置关联进程的总体优先级类别
PrivateMemorySize	获取专用内存大小

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	说　明
PrivateMemorySize64	获取为关联的进程分配的专用内存量
PrivilegedProcessorTime	获取此进程的特权处理器时间
ProcessName	获取该进程的名称
ProcessorAffinity	获取或设置可安排此进程中的线程在其上运行的处理器
Responding	获取指示进程的用户界面当前是否响应的值
SessionId	获取关联的进程的终端服务会话标识符
StandardError	获取用于读取应用程序错误输出的流
StandardInput	获取用于写入应用程序输入的流
StandardOutput	获取用于读取应用程序输出的流
StartInfo	获取或设置要传递给 Process 的 Start 方法的属性
StartTime	获取关联进程启动的时间
SynchronizingObject	获取或设置用于封送由于进程退出事件而发出的事件处理程序调用的对象
Threads	获取在关联进程中运行的一组线程
TotalProcessorTime	获取此进程的总的处理器时间
UserProcessorTime	获取此进程的用户处理器时间
VirtualMemorySize	获取进程的虚拟内存大小
VirtualMemorySize64	获取为关联的进程分配的虚拟内存量
WorkingSet	获取关联进程的物理内存使用情况
WorkingSet64	获取为关联的进程分配的物理内存量



方法

Process 类的常用方法及说明如表 P.20 所示。

表 P.20　Process 类的常用方法及说明

方　法	说　明
BeginErrorReadLine	在应用程序的重定向 StandardError 流上开始进行异步读取操作
BeginOutputReadLine	在应用程序的重定向 StandardOutput 流上开始异步读取操作
CancelErrorRead	取消在应用程序的重定向 StandardError 流上执行的异步读取操作
CancelOutputRead	取消在应用程序的重定向 StandardOutput 流上的异步读取操作
Close	释放与此组件关联的所有资源
CloseMainWindow	通过向进程的主窗口发送关闭消息来关闭拥有用户界面的进程
EnterDebugMode	通过启用当前线程的本机属性 SeDebugPrivilege，将 Process 组件置于与以特殊模式运行的操作系统进程交互的状态
GetProcesses	创建新的 Process 组件的数组，并将它们与现有进程资源关联
Kill	立即停止关联的进程

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	说　明
LeaveDebugMode	使 Process 组件离开允许它与以特殊模式运行的操作系统进程交互的状态
Refresh	放弃有关关联进程的、已缓存到该进程组件内的任何信息
Start	启动进程资源并将其与 Process 组件关联
ToString	如果适用，则将进程的名称格式化为字符串，并与父组件的类型组合
WaitForExit	设置等待关联进程退出的时间，并在该段时间结束前或该进程退出前，阻止当前线程执行
WaitForInputIdle	使 Process 组件等待关联进程进入空闲状态

ProgressBar 控件

ProgressBar 控件通过在水平条中显示相应数目的矩形来指示操作的进度。操作完成时，进度栏被填满。进度栏通常用于帮助用户了解等待一项长时间的操作（例如，加载大文件等）完成所需的时间。

属性

ProgressBar 控件常用属性及说明如表 P.21 所示。

表 P.21　ProgressBar 控件常用的属性及说明

属　性	说　明
Value	此属性表示操作过程中已完成的进度
Maximum	此属性设置进度栏可以显示的最大值
Minimum	此属性设置进度栏可以显示的最小值
Step	此属性可以用于指定 Value 属性递增的值

方法

ProgressBar 控件常用方法及说明如表 P.22 所示。

表 P.22　ProgressBar 控件常用的方法及说明

方　法	说　明
PointToClient	将指定屏幕点的位置计算成工作区坐标
PerformStep	按照 Step 属性的数量增加进度栏的当前位置
Increment	该方法将进度栏的当前位置移动到指定的增量值

下面对比较重要的方法进行详细介绍。

更多资源请访问稀酷客(www.ckook.com)

（1）Increment 方法

该方法将进度栏的当前位置移动到指定的增量值。

语法：

```
public void Increment (int value)
```

Increment 方法的语法中各参数的说明如表 P.23 所示。

表 P.23　Increment 方法的语法中各参数的说明

参　数	描　述
value	增加进度栏的当前位置所根据的数量

（2）PerformStep 方法

该方法按照 Step 属性中指定的数量增加进度栏的值。可以使用 Step 属性指定操作中每个完成的任务更改进度栏的值的数量。

语法：

```
public void PerformStep ()
```

事件

ProgressBar 控件常用事件及说明如表 P.24 所示。

表 P.24　ProgressBar 控件常用的事件及说明

事　件	说　明
Click	在单击控件时发生
GotFocus	在控件接收焦点时发生
DoubleClick	当用户双击控件时发生

Ping 类

Ping 类允许应用程序确定是否可通过网络访问远程计算机。

方法

Ping 类的常用方法及说明如表 P.25 所示。

表 P.25　Ping 类的常用方法及说明

方　法	说　明
Send	尝试将 Internet 控制消息协议（ICMP）回送消息发送到远程计算机并接收来自远程计算机的相应 ICMP 回送答复消息
SendAsync	尝试以异步方式将 Internet 控制消息协议（ICMP）回送消息发送到指定的计算机，并接收来自该计算机的相应 ICMP 回送答复消息

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	说　明
SendAsyncCancel	取消所有挂起的发送 Internet 控制消息协议（ICMP）回送消息并接收相应 ICMP 回送答复消息的异步请求

主要介绍一下 Send 方法，该方法主要用于尝试将 Internet 控制消息协议（ICMP）回送消息发送到远程计算机并接收来自远程计算机的相应 ICMP 回送答复消息。

语法：

```
    public PingReply Send (IPAddress address)
    public PingReply Send (string hostNameOrAddress)
    public PingReply Send (IPAddress address,int timeout)
    public PingReply Send (string hostNameOrAddress,int timeout)
    public PingReply Send (IPAddress address,int timeout,byte[] buffer)
    public PingReply Send (string hostNameOrAddress,int timeout,byte[]
buffer)
    public PingReply Send (IPAddress address,int timeout,byte[]
buffer,PingOptions options)
    public PingReply Send (string hostNameOrAddress,int timeout,byte[]
buffer,PingOptions options)
```

Send 方法的语法中各参数的说明如表 P.26 所示。

表 P.26　Send 方法的语法中各参数的说明

参　数	描　述
address	标识 ICMP 回送消息目标计算机的 IPAddress
HostNameOrAddress	一个字符串类型，它标识作为 ICMP 回送消息目标的计算机。为此参数指定的值可以是主机名，也可以是以字符串形式表示的 IP 地址
timeout	一个 Int32 值，指定（发送回送消息后）等待 ICMP 回送答复消息的最大毫秒数
buffer	一个 Byte 数组，它包含要与 ICMP 回送消息一起发送并在 ICMP 回送应答消息中返回的数据。该数组包含的字节数不能超过 65 500 个字节
options	一个 PingOptions 对象，用于控制 ICMP 回显消息数据包的碎片和生存时间值
返回值	一个 PingReply 对象，如果收到了 ICMP 回显回复消息，则该对象提供相应消息的相关信息；如果未收到消息，则该对象提供失败原因

示例

本示例使用 Ping 类的公共方法 Send()检测远程计算机，在.NET 中 Ping

更多资源请访问稀酷客(www.ckook.com)

类允许应用程序确定是否可通过网络访问远程计算机，示例运行结果如图P.10所示。

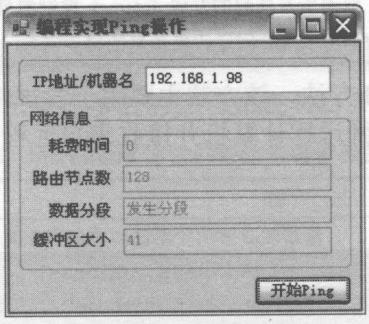


图 P.10　编程实现 Ping 操作

主要代码如下：

```
    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            Ping PingInfo = new Ping();
            PingOptions PingOpt = new PingOptions();
            PingOpt.DontFragment = true;
            string myInfo = "hyworkhyworkhyworkhyworkhyworkhywork";//
定义测试数据
            byte[] bufferInfo = Encoding.ASCII.GetBytes(myInfo);
            int TimeOut = 120;
          PingReply reply = PingInfo.Send(this.textBox1.Text, TimeOut,
bufferInfo, PingOpt);
            if (reply.Status == IPStatus.Success)
            {
                this.textBox2.Text = reply.RoundtripTime.ToString();
                this.textBox3.Text = reply.Options.Ttl.ToString();
           this.textBox4.Text = (reply.Options.DontFragment ? "发生分段" :
"没有发生分段");
                 this.textBox5.Text = reply.Buffer.Length.ToString();
              }
              else
              {
                 MessageBox.Show("无法 Ping 通");
              }
          }
          catch (Exception ey)
          {
```


更多资源请访问稀酷客(www.ckook.com)

```
            MessageBox.Show(ey.Message);
        }
    }
```

PrinterSettings 类

PrinterSettings 类用来指定有关文档打印方式的信息，其中包括打印文档的打印机。

属性

PrinterSettings 类的常用属性及说明如表 P.27 所示。

表 P.27　PrinterSettings 类的常用属性及说明

属　性	说　明
CanDuplex	获取指示打印机是否支持双面打印的值
Collate	获取或设置一个值，该值指示打印文档是否逐份打印
Copies	获取或设置要打印的文档份数
DefaultPageSettings	获取此打印机的默认页设置
Duplex	获取或设置双面打印的打印机设置
FromPage	获取或设置要打印的第一页的页码
IsDefaultPrinter	获取一个值，该值指示 PrinterName 属性是否指定默认打印机
IsValid	获取一个值，该值指示 PrinterName 属性是否指定了有效的打印机
InstalledPrinters	获取安装在计算机上所有打印机的名
LandscapeAngle	获取旋转纵向方向以生成横向方向的角度，以度为单位
MaximumCopies	获取打印机允许用户一次打印的最大份数
MaximumPage	获取或设置可以在 PrintDialog 中选定的最大的 FromPage 或 ToPage
MinimumPage	获取或设置可以在 PrintDialog 中选定的最小的 FromPage 或 ToPage
PaperSizes	获取该打印机支持的纸张大小
PaperSources	获取打印机上可用的纸张来源纸盒
PrinterName	获取或设置要使用的打印机的名称
PrinterResolutions	获取该打印机支持的所有分辨率
PrintFileName	打印到文件时，获取或设置文件名
PrintRange	获取或设置用户已指定要打印的页码
PrintToFile	获取或设置一个值，该值指示打印输出是否发送到文件，而不发送到端口
SupportsColor	获取指示该打印机是否支持彩色打印的值
ToPage	获取或设置要打印的最后一页的页码

更多资源请访问稀酷客(www.ckook.com)

方法

PrinterSettings 类的常用方法及说明如表 P.28 所示。

表 P.28　PrinterSettings 类的常用方法及说明

方　法	说　明
GetHdevmode	创建与打印机设置相对应的 DEVMODE 结构的句柄
GetHdevnames	创建与打印机设置相对应的 DEVNAMES 结构的句柄
IsDirectPrintingSupported	返回一个值，该值指示打印机是否支持打印图像文件
SetHdevmode	将相关信息从给定句柄复制到 PrinterSettings 中
SetHdevnames	将相关信息从给定句柄复制到 PrinterSettings 中

（1）GetHdevmode 方法

创建一个 DEVMODE 结构的句柄，它对应于通过 pageSettings 参数指定的打印机和页设置。

语法：

```
public IntPtr GetHdevmode (PageSettings pageSettings)
```

GetHdevmode 方法的语法中各参数的说明如表 P.29 所示。

表 P.29　GetHdevmode 方法的语法中各参数的说明

参　数	描　述
pageSettings	DEVMODE 结构的句柄所对应于的 PageSettings 对象
返回值	DEVMODE 结构的句柄

（2）IsDirectPrintingSupported 方法

返回一个值，该值指示打印机是否支持打印图像文件。

语法：

```
public bool IsDirectPrintingSupported (Image image)
```

GetHdevmode 方法的语法中各参数的说明如表 P.30 所示。

表 P.30　GetHdevmode 方法的语法中各参数的说明

参　数	描　述
image	要打印的图像
返回值	若打印机支持打印指定的图像，则为 true；否则为 false

更多资源请访问稀酷客(www.ckook.com)

Queue 类

Queue（队列）类主要实现了一个 FIFO（First In First Out，先进先出）的机制。元素在队列的尾部插入（入队操作），并从队列的头部移出（出队操作）。在 Queue 中主要使用 Enqueue、Dequeue、Peek 三种方法对队进行操作。Enqueue 方法用于将对象添加到 Queue 的结尾处，Dequeue 方法移除。

属性

Queue 类的常用属性及说明如表 Q.1 所示。

表 Q.1 Queue 类的常用属性及说明

属　性	说　明
Count	获取 Queue 中包含的元素数
IsSynchronized	获取一个值，该值指示是否同步对 Queue 的访问
SyncRoot	获取可用于同步 Queue 访问的对象

方法

Queue 类的常用方法及说明如表 Q.2 所示。

表 Q.2 Queue 类的常用方法及说明

方　法	说　明
Clear	从 Queue 中移除所有对象
Clone	创建 Queue 的浅表副本
Contains	确定某元素是否在 Queue 中
CopyTo	从指定数组索引开始将 Queue 元素复制到现有一维 Array 中
Dequeue	移除并返回位于 Queue 开始处的对象
Enqueue	将对象添加到 Queue 的结尾处
GetEnumerator	返回循环访问 Queue 的枚举数
Peek	返回位于 Queue 开始处的对象但不将其移除
Synchronized	返回同步的（线程安全）Queue 包装
ToArray	将 Queue 元素复制到新数组
TrimToSize	将容量设置为 Queue 中元素的实际数目

更多资源请访问稀酷客(www.ckook.com)

R

RadioButton 控件

RadioButton 控件，即单选按钮，它为用户提供由两个或多个互斥选项组成的选项集。虽然单选按钮和复选框的功能看似类似，它们却存在重要差异，当用户选择某单选按钮时，同一组中的其他单选按钮不能同时选定。

属性

RadioButton 控件的常用属性及描述如表 R.1 所示。

表 R.1　RadioButton 控件的常用属性及描述

属　性	描　述
BackgroundImage	获取或设置在控件中显示的背景图像
BackgroundImageLayout	获取或设置 ImageLayout 枚举中定义的背景图像布局
BindingContext	获取或设置控件的 BindingContext
CanFocus	获取一个值，该值指示控件是否可以接收焦点
CanSelect	获取一个值，该值指示是否可以选中控件
CheckAlign	获取或设置 RadioButton 的复选框部分的位置
Checked	获取或设置一个值，该值指示是否已选中控件
ContextMenu	获取或设置与控件关联的快捷菜单
FlatAppearance	获取用于指示选中状态和鼠标状态的边框外观和颜色
FlatStyle	获取或设置按钮控件的平面样式外观
Image	获取或设置显示在按钮控件上的图像
ImageList	获取或设置包含按钮控件上显示的 Image 的 ImageList

方法

RadioButton 控件的常用方法及描述如表 R.2 所示。

表 R.2　RadioButton 控件的常用方法及描述

方　法	描　述
FindForm	检索控件所在的窗体
Focus	为控件设置输入焦点
Hide	对用户隐藏控件
PointToClient	将指定屏幕点的位置计算成工作区坐标
PointToScreen	将指定工作区点的位置计算成屏幕坐标

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
ResetText	将 Text 属性重置为其默认值
ResumeLayout	已重载。恢复正常的布局逻辑
Select	已重载。激活控件
SelectNextControl	激活下一个控件
Show	向用户显示控件

事件

RadioButton 控件的常用事件及描述如表 R.3 所示。

表 R.3　RadioButton 控件的常用事件及描述

事　件	描　述
CheckedChanged	当 Checked 属性的值更改时发生
Click	在单击控件时发生
DockChanged	当 Dock 属性的值更改时发生
DoubleClick	当用户双击 RadioButton 控件时发生
TextChanged	在 Text 属性值更改时发生
VisibleChanged	在 Visible 属性值更改时发生

Random 类

表示伪随机数生成器，一种能够产生满足某些随机性统计要求的数字序列的设备。

方法

Random 类的常用方法及描述如表 R.4 所示。

表 R.4　Random 类的常用方法及描述

方　法	描　述
Next	已重载。返回随机数
NextBytes	用随机数填充指定字节数组的元素
NextDouble	返回一个介于 0.0 和 1.0 之间的随机数

示例

本示例向字节数组中添加随机数。

```
byte[] V_b = new byte[1024];
```


更多资源请访问稀酷客(www.ckook.com)

```
Random autoRand = new Random();
autoRand.NextBytes(V_b);
```

本示例用 Random 类获取一个 0.0 到 1.0（不含 1.0）之间的随机数（结果保留一位小数）。

```
private void button2_Click(object sender, EventArgs e)
{
    double V_d = 0.0;
    Random autoRand = new Random();
    V_d = autoRand.NextDouble();
    V_d = Math.Round( V_d,1);
    MessageBox.Show(V_d.ToString());
}
```

本示例使用 Random 类获取与时间相关的随机数、10 以内的随机数、100 到 200 之间的随机数。

```
private void button1_Click(object sender, EventArgs e)
{
    int V_1 = 0;
    int V_2 = 0;
    int V_3 = 0;
    Random autoRand = new Random();
    V_1 = autoRand.Next();
    V_2 = autoRand.Next(10);
    V_3 = autoRand.Next(100, 200);
    MessageBox.Show(V_1.ToString() + '\n' + V_2.ToString() + '\n' +
V_3.ToString());
}
```

Read 方法



语法

- MemoryStream 类的 Read 方法

```
//将二进制流存储到内存流中
MemoryStream stream = new MemoryStream(data)
//在当前流中读取字节块，并写入 buffer 中
Stream.Read(byte[] buffer, int offset, int count);
```



说明

从当前流中读取字节块并将数据写入 buffer 中。

更多资源请访问稀酷客(www.ckook.com)

Read 方法的语法中参数的说明如表 R.5 所示。

表 R.5 Read 方法的语法中参数的说明

参　数	描　述
buffer	存储在当前流中读取的字节块
offset	读取字节块的起始位置
count	读取字节块长度，字节块的最大长度为 1024

- SqlDataReader 类的 Read 方法

```
public override bool Read()
```



说明

使 SqlDataReader 前进到下一条记录。

返回值：如果存在多个行，则为 true；否则为 false。

- Console 类的 Read 方法

```
public static int Read()
```



说明

从标准输入流读取下一个字符。

返回值：输入流中的下一个字符；如果当前没有更多的字符可供读取，则为-1。



命名空间

MemoryStream 类的命名空间为：System.IO

SqlDataReader 类的命名空间为：System.Data.SqlClient

ReadBytes 方法

从当前流中将 count 个字节读入字节数组，并使当前位置提升 count 个字节。



语法

```
public virtual byte[] ReadBytes(int count)
```


更多资源请访问稀酷客(www.ckook.com)

返回值：包含从基础流中读取的数据的字节数组。如果到达了流的末尾，则该字节数组可能小于所请求的字节数。

ReadBytes 方法的语法中参数的说明如表 R.6 所示。

表 R.6　ReadBytes 方法的语法中参数的说明

参　数	描　述
count	要读取的字节数



命名空间

System.IO

ReadLine 方法

从当前流中读取一行字符并将数据作为字符串返回。



语法

```
public override string ReadLine ()
```

返回值：string，输入流中的下一行；如果到达了输入流的末尾，则为空引用。



命名空间

System.IO

ReadOnly 属性

获取或设置一个值，该值指示此 PerformanceCounter 实例是否处于只读模式。



语法

```
public bool ReadOnly { get; set; }
```

属性值：如果 PerformanceCounter 实例为只读模式（即使计数器本身为自定义.NET Framework 计数器），则为 true；如果它为读/写模式，则为 false。默认值为构造函数设置的值。



示例

本示例通过设置 DataGridView 控件的 ReadOnly 属性值为 true，防止用

更多资源请访问稀酷客(www.ckook.com)

户编辑 DataGridView 控件的单元格数据。

```
dataGridView1.ReadOnly = true;
```

本示例是用 ReadOnly 属性设置 RichTextBox 控件为只读。

```
richTextBox1.ReadOnly = true;
```

本示例是用 ReadOnly 属性设置 TextBox 控件为只读。

```
textBox1.ReadOnly = true;
```

ReadToEnd 方法

从流的当前位置到末尾读取流。

语法

```
public override string ReadToEnd ()
```

返回值：String 类型，字符串形式的流的其余部分（从当前位置到末尾）。如果当前位置位于流的末尾，则返回空字符串（“”）。

命名空间

System.IO

示例

本示例使用 StreamReader 类的 ReadToEnd 方法从文本文件中读取所有字符串。

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        openFileDialog1.Filter = "文本文件(*.txt)|*.txt";
        openFileDialog1.ShowDialog();
        textBox1.Text = openFileDialog1.FileName;
        StreamReader SReader = new StreamReader(textBox1.Text,
Encoding.Default);
        textBox2.Text = SReader.ReadToEnd();
          SReader.Close();
    }
    catch { }
}
```


更多资源请访问稀酷客(www.ckook.com)

注意

StreamReader 类对象在使用完毕后，应及时调用 Close()方法关闭。

运行结果如图 R.1 所示。



图 R.1　ReadToEnd 方法应用

ReadXml 函数

使用指定的文件将 XML 架构和数据读入 DataSet。

语法

```
public XmlReadMode ReadXml (string fileName)
```

返回值：作为读取源的文件名（包括路径）。

ReadXml 方法的语法中参数的说明如表 R.7 所示。

表 R.7　ReadXml 方法的语法中参数的说明

参　数	描　述
fileName	作为读取源的文件名（包括路径）

命名空间

System.Data

Receive 方法

返回已由远程主机发送的 UDP 数据报。

语法

```
public byte[] Receive (ref IPEndPoint remoteEP)
```


更多资源请访问稀酷客(www.ckook.com)

返回值：一个类型为 Byte 的数组，它包含数据报数据。

Receive 方法的语法中参数的说明如表 R.8 所示。

表 R.8　Receive 方法的语法中参数的说明

参　数	描　述
remoteEP	一个 IPEndPoint，它表示从其发送数据的远程主机

命名空间

System.Net.Sockets

示例

如果指定了远程计算机发送到本地计算机的端口号，可以通过监听本地端口号来实现对数据的获取，下面就是通过监听本地计算机的端口号“11000”来获取信息的相关代码：

```
UdpClient server = new UdpClient();
IPAddress IP = IPAddress.Parse("127.0.0.1");
IPEndPoint receivePoint = new IPEndPoint(IP, 11000);
byte[] recData = server.Receive(ref receivePoint);
```

Refresh 函数

此方法重新加载当前的网页。

语法

```
public override void Refresh()
```

命名空间

System.Windows.Forms

Regex 类

表示不可变的正则表达式。

命名空间

System.Text.RegularExpressions

更多资源请访问稀酷客(www.ckook.com)

属性

Regex 类的常用属性及描述如表 R.9 所示。

表 R.9 Regex 类的常用属性及描述

属　性	描　述
CacheSize	获取或设置已编译正则表达式的当前 Regex 对象的缓存中的最大项数
Options	返回传入 Regex 构造函数的选项
RightToLeft	获取一个值，该值指示正则表达式是否从右向左进行搜索

方法

Regex 类的常用方法及描述如表 R.10 所示。

表 R.10 Regex 类的常用方法及描述

方　法	描　述
CompileToAssembly	已重载。编译正则表达式，并将其保存到单个程序集的磁盘中
Escape	通过替换为转义码来转义最小的元字符集（\、*、+、?、|、{、[、(、)、^、$、.、# 和空白]
GetGroupNames	返回正则表达式的捕获组名数组
GetGroupNumbers	返回与数组中的组名相对应的捕获组号的数组
GroupNameFromNumber	获取与指定组号相对应的组名
GroupNumberFromName	返回与指定组名相对应的组号
IsMatch	已重载。指示正则表达式在输入字符串中是否找到匹配项
Match	已重载。在输入字符串中搜索正则表达式的匹配项，并将精确结果作为单个 Match 对象返回
Matches	已重载。在输入字符串中搜索正则表达式的所有匹配项并返回所有成功的匹配，就像多次调用 Match 一样
Replace	已重载。 用指定的替换字符串替换由正则表达式定义的字符模式的所有匹配项
Split	已重载。在由正则表达式匹配项定义的位置将输入字符串拆分为一个子字符串数组
Unescape	取消转义输入字符串中的任何转义字符

Registry 类

提供表示 Windows 注册表中的根项的 RegistryKey 对象，并提供访问项/值对的 static 方法。

更多资源请访问稀酷客(www.ckook.com)

命名空间

Microsoft.Win32

方法

Registry 类的常用方法及描述如表 R.11 所示。

表 R.11　Registry 类的常用方法及描述

方　法	描　述
GetValue	检索与指定的注册表项中的指定名称关联的值。如果在指定的项中未找到该名称，则返回您提供的默认值；或者，如果指定的项不存在，则返回空引用
SetValue	已重载。设置注册表项中的名称/值对的值

RegistryKey 类

表示 Windows 注册表中的项级节点。此类是注册表封装。

语法

```
[ComVisibleAttribute(true)]
public sealed class RegistryKey : MarshalByRefObject, Idisposable
```

命名空间

Microsoft.Win32

属性

RegistryKey 类的常用属性及描述如表 R.12 所示。

表 R.12　RegistryKey 类的常用属性及描述

属　性	描　述
Name	检索项的名称
SubKeyCount	检索当前项的子项数目
ValueCount	检索项中值的计数

方法

RegistryKey 类的常用方法及描述如表 R.13 所示。

更多资源请访问稀酷客(www.ckook.com)

表 R.13　RegistryKey 类的常用方法及描述

方　法	描　述
Close	关闭该项，如果该项的内容已修改，则将该项刷新到磁盘
DeleteSubKey	已重载。删除指定的子项。字符串 subkey 不区分大小写
DeleteSubKeyTree	递归删除子项和任何子级子项。字符串 subkey 不区分大小写
DeleteValue	已重载。从此项中删除指定值
Flush	将指定的打开注册表项的全部属性写到注册表中
GetAccessControl	已重载。返回当前注册表项的访问控制安全性
GetSubKeyNames	检索包含所有子项名称的字符串数组
GetValue	已重载。检索与指定名称关联的值
GetValueKind	检索与指定名称关联的值的注册表数据类型
GetValueNames	检索包含与此项关联的所有值名称的字符串数组
OpenRemoteBaseKey	打开一个新的 RegistryKey，它表示远程计算机上的请求的项
OpenSubKey	已重载。检索指定的子项
ReferenceEquals	确定指定的 Object 实例是否是相同的实例
SetAccessControl	向现有注册表项应用 Windows 访问控制安全性
SetValue	已重载。设置注册表项中的名称/值对的值。从所存储数据的类型或指定的 RegistryValueKind 确定注册表数据类型，具体取决于重载



典型应用

本实例是在注册表的 HKEY_CURRENT_USER 根目录下创建并删除指定的文件夹，如图 R.2 所示。



图 R.2　通过 RegistryKey 类来创建注册表文件夹

实现关键代码如下：

```
public static RegistryKey retkey;
private void Form1_Load(object sender, EventArgs e)
{
    //获取 HKEY_CURRENT_USER 根目录节点信息
    retkey = Microsoft.Win32.Registry.CurrentUser;
}
private void button1_Click(object sender, EventArgs e)
{
```


更多资源请访问稀酷客(www.ckook.com)

```
        //在 HKEY_CURRENT_USER 根目录下创建 mrABCD 文件夹
        retkey.CreateSubKey("mrABCD");
    }
    private void button2_Click(object sender, EventArgs e)
    {
        //在 HKEY_CURRENT_USER 根目录下删除 mrABCD 文件夹
        retkey.DeleteSubKey("mrABCD");
    }
```

Remove 方法

语法

- String 类的 Remove 方法

```
Public String Remove ( int startIndex, int clunt);
```

说明

从一个字符串的指定位置开始，删除指定数量的字符。

返回值：一个新的 String，它等于此实例减去 count 数目的字符。

Remove 方法的语法各参数的说明如表 R.14 所示。

表 R.14　Remove 方法语法各参数的说明

参　数	描　述
startIndex	用于指定开始删除的位置，索引从 0 开始
clunt	指定删除的字符数量

- BindingSource 类的 Remove 方法

```
public virtual void Remove(Object value)
```

说明

从列表中移除指定的项。

Remove 方法的语法各参数的说明如表 R.15 所示。

表 R.15　Remove 方法语法各参数的说明

参　数	描　述
value	由 List 属性表示的基础列表中移除的项

更多资源请访问稀酷客(www.ckook.com)

典型应用

下面使用 Remove 方法移除 ListBox 控件中的选中项，如图 R.3 所示。



图 R.3　移除选中的项

实现相关代码如下：

```
private void button1_Click(object sender, EventArgs e)
{
    if (listBox1.Items.Count > 0)//判断 listBox1 控件中是否有数据项
    {
        if (listBox1.SelectedItems.Count > 0)//判断是否有选中项
            //移除指定的选中项
            listBox1.Items.Remove(listBox1.SelectedItem.ToString());
        else
            return;
    }
}
```

RemoveAt 方法

移除此列表中指定索引处的项。

语法

```
public virtual void RemoveAt (int index)
```

RemoveAt 方法的语法中各参数的说明如表 R.16 所示。

表 R.16　RemoveAt 方法语法中各参数的说明

参　数	描　述
index	从 0 开始的索引（属于要移除的项）

示例

本示例使用 RemoveAt 方法移除 ArrayList 类中的项。

更多资源请访问稀酷客(www.ckook.com)

```
ArrayList myAL = new ArrayList();
myAL.Add("The");
myAL.Add("quick");
myAL.Add("brown");
myAL.RemoveAt(1);
```

本示例使用 RemoveAt 方法移除 ListBox 控件中被选中的项。

```
private void button1_Click(object sender, EventArgs e)
{
    if (listBox1.SelectedIndex >= 0)//判断是否有选中的项
        //根据选中项的索引号移除指定的项
        listBox1.Items.RemoveAt(listBox1.SelectedIndex);
}
```


Replace 方法

语法

- File 类的 Replace 方法

```
public static void Replace (string sourceFileName, string
destinationFileName, string destinationBackupFileName)
```

说明

使用其他文件的内容替换指定文件的内容，这一过程将删除原始文件，并创建被替换文件的备份。

Replace 方法的语法各参数的说明如表 R.17 所示。

表 R.17 Replace 方法语法各参数的说明

参　数	描　述
sourceFileName	替换由 destinationFileName 指定的文件的文件名
destinationFileName	替换文件的名称
destinationBackupFileName	备份文件的名称

- Regex 类的 Replace 方法

```
public string Replace (string input,MatchEvaluator evaluator)
Replace(string input,string replacement)
Replace(string input,string replacement,int count)
Replace(string input,string replacement,int count,int startat)
```


更多资源请访问稀酷客(www.ckook.com)

说明

用指定的替换字符串替换由正则表达式定义的字符模式的所有匹配项。

返回值：String 类型，已修改的字符串。

Replace 方法的语法各参数的说明如表 R.18 所示。

表 R.18 Replace 方法语法各参数的说明

参　数	描　述
input	要修改的字符串
evaluator	在每一步计算替换的 MatchEvaluator
replacement	替换字符串
count	可进行替换的最大次数
startat	输入字符串中开始搜索的字符位置

示例

本示例使用 File 类的 Replace 方法将 test.txt 文件中的内容替换成 test2.txt 文件中的内容，并删除 test.txt 文件，再根据 test.txt 文件中的内容创建一个备份文件 test3.text.bac。

```
    private void button1_Click(object sender, EventArgs e)
    {
        File.Replace(@"C:\test.txt", @"C:\test2.txt", @"C:\test3.text.
bac");
    }
```

注意

test.txt 和 test2.txt 文件必须存在。

本示例使用 Regex 类的 Replace 方法从要修改字符串的第一个字符开始，用替换字符串替换由“[A-Za-z]*?”表达式所匹配的所有匹配项。

```
    private void button1_Click(object sender, EventArgs e)
    {
        string strResult = System.Text.RegularExpressions.Regex.
    Replace(textBox1.Text, @"[A-Za-z]\*?", textBox2.Text);
        MessageBox.Show("替换前字符:" + "\n" + textBox1.Text + "\n" + "
```


更多资源请访问稀酷客(www.ckook.com)

```
替换的字符:" + "\n" + textBox2.Text + "\n" + "替换后的字符:" + "\n"
 + strResult,"替换");
}
```

运行结果如图 R.4 所示。

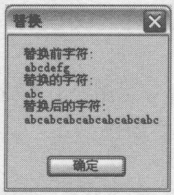


图 R.4　替换匹配的字符串

本示例使用 Regex 类的 Replace 方法用“,”替换字符串中的空格。

```
private void button2_Click(object sender, EventArgs e)
{
    Regex rgx = new Regex(@"\s+");
    MessageBox.Show(rgx.Replace("a b c d", ","));
}
```

运行结果：a,b,c,d

本示例使用 Regex 类的 Replace 方法用“,”替换字符串中的空格，只替换两次。

```
private void button2_Click(object sender, EventArgs e)
{
    Regex rgx = new Regex(@"\s+");//定义正则表达式
    MessageBox.Show(rgx.Replace("a b c d", ",", 2));
}
```

运行结果：a,b,c d

Return 语句

功能

用于终止它所在方法的执行，并将控制返回给调用方法。它还可以返回一个可选值，该值可以为任何类型变量、结果集等。灵活使用该语句可以使程序工作效率更高。

示例

使用 Return 语句按一定条件结束死循环。

更多资源请访问稀酷客(www.ckook.com)

```
private void button1_Click(object sender, EventArgs e)
{
    int var_i = 0;
    while (true)
    {
        MessageBox.Show(var_i.ToString());
        var_i++;
        if (var_i == 10)//如果循环了 10 次
            return;//退出本次循环
    }
}
```

使用 Return 语句自定义一个有返回值的方法。

```
public string SetReturn(string Str)//自定义一个有返回值的方法
{
    return "当前返回的值是："+Str;
}
private void button2_Click(object sender, EventArgs e)
{
    MessageBox.Show(SetReturn("mr"));//调用自定义方法
}
```

RichTextBox 控件

RichTextBox 控件用于显示、输入和操作带有格式的文本。该控件除了能够执行 TextBox 控件的所有功能之外，它还有如下功能：显示字体、颜色、超级链接、从文件加载文本或者图像、撤销、重复、编辑操作以及查找指定的字符。RichTextBox 通常用于提供文本操作和显示功能。与 TextBox 控件一样，该控件也可以显示滚动条，但与 TextBox 控件不同的是，默认情况下，该控件将同时显示水平滚动条和垂直滚动条，并具有更多的滚动条设置。

属性

RichTextBox 控件常用属性及说明如表 R.19 所示。

表 R.19　RichTextBox 控件常用属性及说明

属　性	描　述
DetectUrls	此属性用于指定键入 URL 时，RTF 文本框执行的操作。当此属性设为 Fasle 时，RTF 文本框将 URL 视为普通的文本，当设为 true 时，RTF 文本框将 URL 自动转换为 URL 格式
RedoActionName	此属性表示当调用 Redo 方法时，可再次在 RichTextBox 控件上执行的操作

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
SelecteText	此属性表示在当前 RichTextBox 控件中选定的文本
SelectionColor	此属性表示当前选中的文本和当前插入点文本颜色
ZoomFactor	此属性表示 RichTextBox 控件中内容当前缩放的级别或百分比

方法

RichTextBox 控件常用方法及说明如表 R.20 所示。

表 R.20　RichTextBox 控件常用方法及说明

方　法	描　述
Find	此方法用于在 RichTextBox 控件中搜索指定文本
GetCharFromPosition	此方法用于检索 RTF 文本框中指定位置的字符。如果该位置没有字符，则返回指定位置最近的字符
LoadFile	此方法用于指定文件的内容加载或检索到 RichTextBox 控件中
Redo	此方法用于再次执行已撤销或取消的操作
SaveFile	此方法用于将 RTF 文本框的内容保存到指定文件

事件

RichTextBox 控件常用事件及说明如表 R.21 所示。

表 R.21　RichTextBox 控件常用事件及说明

事　件	描　述
TextChanged	在 Text 属性值更改时发生
Click	在单击控件时发生
SelectionChanged	当在 RichTextBox 中选择文本或修改文本时，触发该事件
LinkClicked	单击出现在 RichTextBox 控件中的链接时，触发该事件

典型应用

下面使用 RichTextBox 控件显示 rtf 格式的文本及图片。如图 R.5 所示。

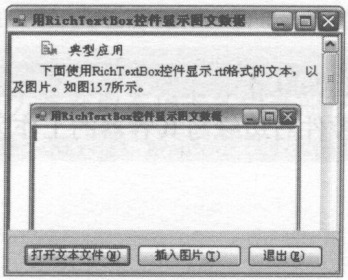


图 R.5　RichTextBox 控件显示文本及图片

更多资源请访问稀酷客(www.ckook.com)

实现相关代码如下：

```
    public void OpenFile()//自定义方法，向 richTextBox1 控件中添加 rtf 文本
    {
        openFileDialog1.Filter = "文本文件（*.txt）|*.txt|rtf 文件（*.rtf）
|*.rtf";
        openFileDialog1.Title = "打开";
        openFileDialog1.FilterIndex = 2;
        if (openFileDialog1.ShowDialog() == DialogResult.OK)
        {
            richTextBox1.Clear();
            richTextBox1.LoadFile(openFileDialog1.FileName,
    RichTextBoxStreamType.RichText);
        }
    }
    public void OpenPicuter()//自定义方法，向 richTextBox1 控件中添加图片
    {
        openFileDialog1.Filter = "bmp 文件（*.bmp）|*.bmp|jpg 文件（*.jpg）
    |*.jpg|ico 文件（*.ico）|*.ico";
         openFileDialog1.Title = "打开图片";
         openFileDialog1.FilterIndex=2;
         if(openFileDialog1.ShowDialog()==DialogResult.OK)
         {
             Bitmap bmp = new Bitmap(openFileDialog1.FileName);
             Clipboard.SetDataObject(bmp,false);
             if (richTextBox1.CanPaste(DataFormats.GetFormat
    (DataFormats.Bitmap)))
                 richTextBox1.Paste();
         }
    }
```

Right 属性

获取控件右边缘与其容器的工作区左边缘之间的距离（以像素为单位）。

语法

```
public int Right { get; }
```

属性值：Int32 表示控件右边缘与其容器的工作区左边缘之间的距离（以像素为单位）。

示例

本示例是设置 Panel 控件右边缘与窗体相距 20。

更多资源请访问稀酷客(www.ckook.com)

```
Panel1.Right = 20;
```

本示例用 Right 属性获取当前控件右边缘与其父容器左边缘之间的大小。

```
MessageBox.Show("Right: " + button1.Right.ToString());
```

Round 方法

将小数值舍入到指定的精度（四舍五入）。

语法

```
public static decimal Round ( decimal d, int decimals)
```

Round 方法的语法中的各参数说明如表 R.22 所示。

表 R.22　Round 方法语法中的各参数的说明

参　数	描　述
d	要舍入的小数
decimals	返回值中的有效数字位数（精度）
返回值	表示 128 位数据类型。同浮点型相比，decimal 类型具有更高的精度和更小的范围，这使它适合于财务和货币计算

RowCount 方法

获取或设置 DataGridView 中显示的行数。

语法

```
public int RowCount { get; set; }
```

返回值：int 型，显示 DataGridView 中的列数。

示例

本示例用 RowCount 方法获取 DataGridView 的当前行数。

```
MessageBox.Show(dataGridView1.RowCount.ToString());
```

本示例用 RowCount 方法获取选项卡条中当前正显示的行数。

```
MessageBox.Show(TabControl1.RowCount.ToString());
```

Rows 属性

获取一个集合，该集合包含 DataGridView 控件中的所有行。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public DataGridViewRowCollection Rows { get; }
```

属性值：一个 DataGridViewRowCollection，包含 DataGridView 控件中的所有行。

示例

本示例用 Rows 属性获取 DataGridView 的当前行数。

```
MessageBox.Show(dataGridView1.Rows.Count.ToString());
```

本示例使用 DataGridView 控件的 Rows 属性向该控中添加数据。运行结果如图 R.6 所示。



图 R.6　向 DataGridView 控件添加数据

实现相关代码如下：

```
    private void Form1_Load(object sender, EventArgs e)
    {
        dataGridView1.ColumnCount = 4; //指定 DataGridView 控件显示的列数
        dataGridView1.ColumnHeadersVisible = true;
        DataGridViewCellStyle columnHeaderStyle = new
    DataGridViewCellStyle();          //设置 DataGridView 控件标题列的样式
        columnHeaderStyle.BackColor = Color.Beige;
        columnHeaderStyle.Font = new Font("Verdana", 10, FontStyle.
Bold);
        dataGridView1.ColumnHeadersDefaultCellStyle = columnHeaderStyle;
        //设置 DataGridView 控件的标题列名
        dataGridView1.Columns[0].Name = "编号";
        dataGridView1.Columns[1].Name = "姓名";
        dataGridView1.Columns[2].Name = "年龄";
        dataGridView1.Columns[3].Name = "性别";
        string[] row1 = new string[] { "0001", "小吕", "28","男" };
        string[] row2 = new string[] { "0002", "小张", "27", "男" };
        string[] row3 = new string[] { "0003", "小郭", "24", "女" };
        string[] row4 = new string[] { "0004", "小贯", "21", "女" };
```


更多资源请访问稀酷客(www.ckook.com)

```
    string[] row5 = new string[] { "0005", "小陈", "20", "女" };
    string[] row6 = new string[] { "0006", "小梁", "23", "男" };
    object[] rows = new object[] { row1, row2, row3, row4, row5, row6 };
    foreach (string[] rowArray in rows)
    {
        dataGridView1.Rows.Add(rowArray);
    }
}
```


更多资源请访问稀酷客(www.ckook.com)

S

Save 方法

将图片以指定格式保存到指定文件中。



语法

```
public void Save (string filename, ImageFormat format)
```

Save 方法的语法中各参数的说明如表 S.1 所示。

表 S.1 Save 方法的语法中各参数的说明

参　数	描　述
filename	字符串，包含要将图片保存到的文件的名称
format	要保存图片的 ImageFormat 格式

SaveFile 方法

将 RichTextBox 中的内容保存到特定类型的文件中。



语法

```
public void SaveFile (string path,RichTextBoxStreamType fileType)
```

SaveFile 方法的语法中各参数的说明如表 S.2 所示。

表 S.2 SaveFile 方法的语法中各参数的说明

参　数	描　述
path	要保存的文件的名称和位置
fileType	RichTextBoxStreamType 值之一

RichTextBoxStreamType 枚举各成员说明如表 S.3 所示。

表 S.3 RichTextBoxStreamType 枚举各成员说明

成　员	描　述
PlainText	用空格代替对象链接与嵌入（OLE）对象的纯文本流
RichNoOleObjs	用空格代替 OLE 对象的丰富文本格式（RTF 格式）流。该值只在用于 RichTextBox 控件的 SaveFile 方法时有效
RichText	RTF 格式流
TextTextOleObjs	具有 OLE 对象的文本表示形式的纯文本流。该值只在用于 RichTextBox 控件的 SaveFile 方法时有效

更多资源请访问稀酷客(www.ckook.com)

续表

成　员	描　述
UnicodePlainText	包含用空格代替对象链接与嵌入(OLE)对象的文本流。该文本采用 Unicode 编码

示例

本示例是将 RichTextBox 控件中的内容以 RTF 格式文件进行保存。

```
richTextBox1.SaveFile("c:\\Txt1.rtf");
```

本示例使用 SaveFile 方法将 RichTextBox 控件中的文本信息以 txt 文件的形式保存到指定的目录下。

```
private void button1_Click(object sender, EventArgs e)
{
    saveFileDialog1.Filter = "文本文件 (*.txt) |*.txt";
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        richTextBox1.SaveFile(saveFileDialog1.FileName,
RichTextBoxStreamType.PlainText);
    }
}
```

SaveFileDialog 组件

显示一个预先配置的对话框，用户可以使用该对话框将文件保存到指定的位置。

属性

SaveFileDialog 组件的常用属性及说明如表 S.4 所示。

表 S.4　SaveFileDialog 组件的常用属性及说明

属　性	描　述
CreatePrompt	获取或设置一个值，该值指示如果用户指定不存在的文件，对话框是否提示用户允许创建该文件
OverwritePrompt	获取或设置一个值，该值指示如果用户指定的文件名已存在，Save As 对话框是否显示警告
FileName	获取或设置一个包含在文件对话框中选定的文件名的字符串
FileNames	获取对话框中所有选定文件的文件名
Filter	获取或设置当前文件名筛选器字符串，该字符串决定对话框的“另存为文件类型”或“文件类型”框中出现的选择内容

更多资源请访问稀酷客(www.ckook.com)

方法

SaveFileDialog 组件的常用方法及说明如表 S.5 所示。

表 S.5　SaveFileDialog 组件的常用方法及说明

方　法	描　述
ShowDialog	已重载。运行通用对话框
OpenFile	打开用户选定的具有读/写权限的文件

事件

SaveFileDialog 组件的常用事件及说明如表 S.6 所示。

表 S.6　SaveFileDialog 组件的常用事件及说明

事　件	描　述
FileOk	当用户单击文件对话框中的“打开”或“保存”按钮时发生
HelpRequest	当用户单击通用对话框中的“帮助”按钮时发生

ScrollBars 属性

语法

- Panel 控件的 ScrollBars 属性

```
public virtual ScrollBars ScrollBars { get; set; }
```

说明

获取或设置 Panel 控件中滚动条的可见性和位置。

- TextBox 控件的 ScrollBars 属性

```
[LocalizableAttribute(true)]
public ScrollBars ScrollBars { get; set; }
```

说明

获取或设置哪些滚动条应出现在多行 TextBox 控件中。

- DataGridView 控件的 ScrollBars 属性

```
[LocalizableAttribute(true)]
```


更多资源请访问稀酷客(www.ckook.com)

```
public ScrollBars ScrollBars { get; set; }
```

说明

获取或设置要在 DataGridView 控件中显示的滚动条的类型。

属性值：ScrollBars 枚举值之一。默认为 None。

ScrollBars 枚举成员的说明如表 S.7 所示。

表 S.7　ScrollBars 枚举成员的说明

成　员	描　述
Auto	根据需要，可显示水平滚动条、垂直滚动条或这两种滚动条。要不然也可以不显示任何滚动条
Both	同时显示水平滚动条和垂直滚动条
Horizontal	只显示水平滚动条
None	不显示滚动条
Vertical	只显示垂直滚动条

示例

本示例使用 ScrollBars 属性设置 textBox1 控件只有垂直滚动条。

```
textBox1.ScrollBars = ScrollBars.Vertical;
```

本示例使用 ScrollBars 属性设置 dataGridView1 控件即有水平滚动条，又有垂直滚动条。

```
dataGridView1.ScrollBars = ScrollBars.Both;
```

本示例使用 ScrollBars 属性设置 pabel1 控件只有水平滚动条。

```
pabel1.ScrollBars = ScrollBars.Horizontal;
```

Select 方法

语法

- Control 类的 Select 方法

```
public void Select ()
protected virtual void Select (    bool directed,    bool
forward)
```


更多资源请访问稀酷客(www.ckook.com)

说明

用于激活控件，带参数的语法可以指定从中选择控件的 Tab 键顺序的方向。

Select 方法的各参数说明如表 S.8 所示。

表 S.8　Select 方法的各参数说明

参　数	描　述
directed	true 表示要指定选择控件的方向；false 表示不指定
forward	如果为 true 则在 Tab 键顺序中前移；如果为 false 则在 Tab 键顺序中后移

- ComboBox 控件的 Select 方法

```
public void Select ( int start,  int length)
```

说明

选择可编辑部分中的文本范围。

Select 方法的各参数说明如表 S.9 所示。

表 S.9　Select 方法的各参数说明

参　数	描　述
start	文本框中当前选定文本的第一个字符的位置
length	要选择的字符数

典型应用

在编写应用程序时，信息的录入是非常必要的，那么，在录入信息时，如何按指定的顺序使文本框获得焦点呢？可以通过对控件的遍历及 Select 方法来实现。如图 S.1 所示。

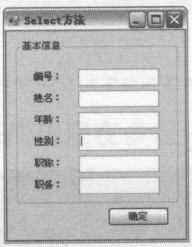


图 S.1　使用 Select 方法使文本框获得焦点

更多资源请访问稀酷客(www.ckook.com)

实现代码如下：

```
//使控件循环获得焦点
public bool KeySelect(int n, Control.ControlCollection Con)
{//n:上一个获得焦点的控件标识;    Con: 控件内的控件集合
    bool var_bool = false;
    foreach (Control C in Con)//遍历控件中的所有控件
    {
        if (C.GetType().Name=="TextBox")//判断当前是否为 TextBox 控件
            //查找下一个获得焦点的控件
            if ((Convert.ToInt16(((TextBox)C).Tag.ToString())) == (n + 1))
            {
                ((TextBox)C).Select();//使控件获得焦点
                var_bool = true;//标识已找到获取焦点的控件
                break;
            }
    }
    return var_bool;
}
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == '\r')//如果键值为回车键
    {
        //如果到了最后一个录入信息的文本框
        if (KeySelect(Convert.ToInt16(((TextBox)sender).Tag.ToString()),
((TextBox)sender).Parent.Controls) == false)
            button1.Select();//将焦点设置在“确定”按钮上
    }
}
```



注意

所有 TextBox 控件及 button1 控件的 KeyPress 事件，都要选择 textBox1_KeyPress。

SelectAll 方法

选定文本框中的所有文本。



语法

```
public void SelectAll()
```


更多资源请访问稀酷客(www.ckook.com)

示例

本示例使用 SelectAll 方法可以选择文本框中的所有文本，如 TextBos、RichTextBox 控件等。

```
if (richTextBox1.SelectionLength == 0)//如果没有被选中的文本
    richTextBox1.SelectAll();//选择控件中的所有文本
richTextBox1.Select();//激活控件，使其显示激活的文本
```

本示例使用 SelectAll 方法选择 ComboBox 可编辑部分中的所有文本。

```
comboBox1.SelectAll();
```

Selected 属性

语法

```
public virtual bool Selected { get; set; }
```

说明

（1）ToolStripItem 类的 Selected 属性，用于获取一个值，该值指示该项是否处于选定状态。

（2）DataGridViewCell 类的 Selected 属性，用于获取或设置一个值，该值指示是否已选定该单元格。

（3）ListViewItem 类的 Selected 属性，用于获取或设置一个值，该值指示是否选定此项。

属性值：如果选定，则为 true；否则为 false。默认为 false。

示例

本示例使用 Selected 属性使 listView1 控件的第一个项为被选中状态。

```
listView1.Items[0].Selected = true;
```

SelectedCells 属性

获取用户选定的单元格的集合。

语法

```
ChrB(charcode)
```


更多资源请访问稀酷客(www.ckook.com)

属性值：一个 DataGridViewSelectedCellCollection，表示用户选定的单元格。

示例

本示例使用 SelectedCells 属性显示 dataGridView1 控件中被选中的所有单元格的值。

```
for (int i = 0; i < dataGridView1.SelectedCells.Count; i++)
{
    MessageBox.Show(dataGridView1.SelectedCells[i].Value.ToString());
}
```

本示例用于选中当前被选中单元格所在的行。

```
dataGridView1.SelectedCells[0].OwningRow.Selected = true;
```

SelectedColumns 属性

获取用户选定的列的集合。

语法

```
public DataGridViewSelectedColumnCollection SelectedColumns
{ get; }op
```

属性值：一个 DataGridViewSelectedColumnCollection，表示用户选定的列。

注意

必须将 SelectionMode 属性设置为 DataGridViewSelectionMode.FullColumnSelect 或 DataGridViewSelectionMode.ColumnHeaderSelect，才能用选定的列填充 SelectedColumns 属性。

SelectedIndex 属性

语法

- TabControl 控件的 SelectedIndex 属性

```
public int SelectedIndex { get; set; }
```


更多资源请访问稀酷客(www.ckook.com)



说明

获取或设置当前选定的选项卡页的索引。

属性值：当前选定的选项卡页的从零开始的索引。默认值为-1，当未选定任何选项卡页时也为此值。

- ListBox 控件的 SelectedIndex 属性

```
[BindableAttribute(true)]
public virtual int SelectedIndex { get; set; }
```



说明

获取或设置 ListBox 中当前选定项的从 0 开始的索引。

属性值：当前选定项的从零开始的索引。如果未选定任何项，则返回值为-1。

SelectedItem 属性

表示当前选中的项。



语法

```
public Object SelectedItem { get; set; }
```

属性值：表示控件中当前选定内容的对象。

SelectedItems 属性

获取一个包含当前在环境中选定的项的集合。



语法

```
public SelectedListViewItemCollection SelectedItems { get; }
```

属性值：ListView.SelectedListViewItemCollection，包含在控件中选定的项。如果当前没有选定的项，则返回空 ListView.SelectedListViewItemCollection。

更多资源请访问稀酷客(www.ckook.com)

示例

本示例用 SelectedItems 属性中的 count 属性获取当前被选中集合的个数。

```
listView1.SelectedItems.Count;
```

SelectedPath 属性

获取或设置用户选定的路径。

语法

```
public string SelectedPath { get; set; }
```

属性值：对话框中选定的第一个文件夹或用户选定的最后一个文件夹的路径。默认值为空字符串。

注意

必须将 SelectionMode 属性设置为 FullRowSelect 或 RowHeaderSelect，才能用选定的行填充 SelectedRows 属性。

示例

本示例使用 SelectedPath 属性获取当前所选文件夹的路径。

```
private void button1_Click(object sender, EventArgs e)
{
    //如果选中文件夹并单击“确定”按钮
    if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)
    {
        label1.Text = "选择文件夹的位置：" + folderBrowserDialog1.
SelectedPath.ToString();
    }
}
```

SelectedRows 属性

获取用户选定的行的集合。

语法

```
public DataGridViewSelectedRowCollection SelectedRows { get; }
```


更多资源请访问稀酷客(www.ckook.com)

属性值：一个 DataGridViewSelectedRowCollection，包含用户选定的行。

注意

必须将 SelectionMode 属性设置为 FullRowSelect 或 RowHeaderSelect，才能用选定的行填充 SelectedRows 属性。

典型应用

在用 DataGridView 控件显示数据表信息后，有时会在 DataGridView 控件中选中多行数据，可以用 SelectedRows 属性来获取多行数据中的各单元格信息。如图 S.2 所示。

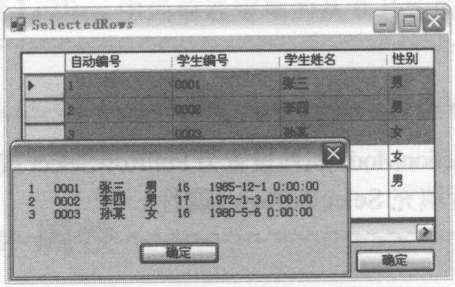


图 S.2　SelectedRows 属性的应用

实现关键代码如下：

```
private void Form1_Load(object sender, EventArgs e)
{
    //用 SqlConnection 对象与指定的数据库相连接
    SqlConnection My_con = new SqlConnection("Data Source=127.0.0.1;
Initial Catalog=db_S;User ID=sa");
    My_con.Open();  //打开数据库连接
    SqlDataAdapter SQLda = new SqlDataAdapter("select * from tb_S1",
My_con);  //创建一个 SqlDataAdapter 对象，并获取指定数据表的信息
    DataSet My_DataSet = new DataSet(); //创建 DataSet 对象
    SQLda.Fill(My_DataSet);//通过 SqlDataAdapter 对象的 Fill()方法，将数
据表信息添加到 DataSet 对象中
    dataGridView1.DataSource = My_DataSet.Tables[0];
    dataGridView1.SelectionMode = DataGridViewSelectionMode
.FullRowSelect;//单击单元格时选定整行
}
private void button1_Click(object sender, EventArgs e)
{
```


更多资源请访问稀酷客(www.ckook.com)

```
    string var_S = "";//定义一个变量
    if (dataGridView1.SelectedRows.Count > 0)//如果有选中的行
    {
        //遍历选中的行
        for (int i = 0; i < dataGridView1.SelectedRows.Count; i++)
        {
            //遍历列
            for (int j = 0; j < dataGridView1.ColumnCount; j++)
                var_S = var_S + dataGridView1.SelectedRows[i].Cells[j].
Value.ToString()+"   ";//记录各列中的信息
            var_S = var_S + "\n\r";//如果选中的多行，换行
        }
        MessageBox.Show(var_S);
    }
}
```

SelectedText 属性

获取或设置可编辑部分中选定的文本。

语法

```
public override string SelectedText { get; set; }
```

属性值：表示组合框中当前选定文本的字符串。如果 DropDownStyle 设置为 DropDownList，则返回值是空字符串("")。

示例

本示例使用 SelectedText 属性显示 ComboBox 控件被选中的文本。

```
MessageBox.Show(comboBox1.SelectedText);
```

本示例使用 SelectedText 属性显示 RichTextBox 控件被选中的文本。

```
MessageBox.Show(richTextBox1.SelectedText);
```

本示例使用 SelectedText 属性判断 TextBox 控件的可编辑区域中是否有被选中的文本，如果有则清空。

```
if (textBox1.SelectedText != "")
    textBox1.SelectedText = "";
```

SelectionColor 属性

获取或设置当前选定文本或插入点的文本颜色。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public Color SelectionColor { get; set; }
```

属性值：表示应用到当前选定文本或在插入点后输入的文本的颜色的Color。

SelectionEnd 属性

获取或设置选定日期范围的结束日期。

语法

```
public DateTime SelectionEnd { get; set; }
```

属性值：DateTime，指示选择范围中的最后日期。

SelectionFont 属性

获取或设置当前选定文本或插入点的字体。

语法

```
public Font SelectionFont { get; set; }
```

属性值：表示应用到当前选定文本或在插入点后输入的文本的字体的Font。

示例

本示例使用 SelectionFont 属性来设置 RichTextBox 控件中被选中文本或插入点的字体。

```
private void button1_Click(object sender, EventArgs e)
{
    if (fontDialog1.ShowDialog() == DialogResult.OK)
    {
        richTextBox1.SelectionFont = fontDialog1.Font;
    }
}
```

运行结果如图 S.3 所示。

更多资源请访问稀酷客(www.ckook.com)

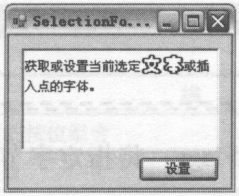


图 S.3　SelectionFont 属性的应用

Send 方法

将数据发送到连接的 Socket。

语法

```
public int Send (byte[] buffer)
```

说明

将数据发送到连接的 Socket。

```
public int Send (IList<ArraySegment<byte>> buffers)
```

说明

将列表中的一组缓冲区发送到连接的 Socket。

```
public int Send (byte[] buffer,    SocketFlags socketFlags)
```

说明

使用指定的 SocketFlags 将数据发送到连接的 Socket。

```
public int Send (IList<ArraySegment<byte>> buffers,SocketFlags
socketFlags)
```

说明

使用指定的 SocketFlags，将列表中的一组缓冲区发送到连接的 Socket。

更多资源请访问稀酷客(www.ckook.com)

```
public int Send (byte[] buffer,int size,SocketFlags socketFlags)
```



说明

使用指定的 SocketFlags，将指定字节数的数据发送到已连接的 Socket。

```
[CLSCompliantAttribute(false)]
public int Send (IList<ArraySegment<byte>> buffers,SocketFlags
socketFlags,out SocketError errorCode)
```



说明

使用指定的 SocketFlags，将列表中的一组缓冲区发送到连接的 Socket。

```
public int Send (byte[] buffer,    int offset,  int
size,SocketFlags socketFlags)
```



说明

使用指定的 SocketFlags，将指定字节数的数据发送到已连接的 Socket（从指定的偏移量开始）。

```
public int Send (byte[] buffer,int offset,  int size,SocketFlags
socketFlags,out SocketError errorCode)
```



说明

从指定的偏移量开始使用指定的 SocketFlags 将指定字节数的数据发送到连接的 Socket。

返回值：已发送到 Socket 的字节数。

Send 方法的各参数说明如表 S.10 所示。

表 S.10　Send 方法的各参数说明

参　数	描　述
buffer	Byte 类型的数组，它包含要发送的数据
buffers	Byte 类型的 ArraySegment 的列表，它包含要发送的数据

更多资源请访问稀酷客(www.ckook.com)

续表

参　数	描　述
SocketFlags	SocketFlags 值的按位组合
size	要发送的字节数
errorCode	一个 SocketError 对象，它存储套接字错误
offset	数据缓冲区中开始发送数据的位置

SetDate 方法

将日期设置为当前选定的日期。

语法

```
public void SetDate (DateTime date)
```

SetDate 方法的各参数说明如表 S.11 所示。

表 S.11　SetDate 方法的各参数说明

参　数	描　述
date	要选择的日期

示例

本示例使用 SetDate 方法在窗体载入时在 MonthCalendar 控件中显示当前日期。

```
private void Form1_Load(object sender, EventArgs e)
{
    DateTime var_dtime = System.DateTime.Now;//获取系统当前时间
    monthCalendar1.SetDate(var_dtime);//显示当前时间
}
```

SetError 方法

将 Error 属性设置为指定的 TextWriter 对象。

语法

```
public static void SetError (TextWriter newError)
```

SetError 方法的各参数说明如表 S.12 所示。

更多资源请访问稀酷客(www.ckook.com)

表 S.12　SetError 方法的各参数说明

参　数	描　述
newError	一个 TextWriter 流，它是新的标准错误输出

示例

本示例使用 ErrorProvider 控件的 SetError 方法在 TextBox 控件的右边显示一个错误标记 。

```
errorProvider1.SetError(textBox1, "登录名错误");
```

SetPixel 方法

获取 Bitmap 图像中指定像素的颜色。

语法

```
public void SetPixel (int x,int y,Color color)
```

SetPixel 方法的各参数说明如表 S.13 所示。

表 S.13　SetPixel 方法的各参数说明

参　数	描　述
x	要设置的像素的 x 坐标
y	要设置的像素的 y 坐标
color	Color 结构，它表示要分配给指定像素的颜色

SetToolTip 方法

使工具提示文本与指定的控件相关联。

语法

```
public void SetToolTip (Control control,string caption)
```

SetToolTip 方法的各参数说明如表 S.14 所示。

表 S.14　SetToolTip 方法的各参数说明

参　数	描　述
control	要将工具提示文本与其关联的 Control
caption	指针位于控件上方时要显示的工具提示文本

更多资源请访问稀酷客(www.ckook.com)

示例

在鼠标移入 TextBox 控件的可见区域时，用 ToolTip 控件的 SetToolTip 方法显示一个提示框，说明该控件用于录入姓名。

```
private void textBox1_MouseEnter(object sender, EventArgs e)
{
    toolTip1.SetToolTip(textBox1, "请录入姓名");
}
```

运行结果如图 S.4 所示。

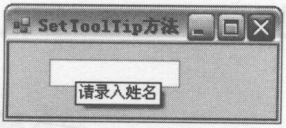


图 S.4　SetToolTip 方法的应用

SetValue 方法

语法

- Registry 对象的 SetValue 方法

```
public static void SetValue (string keyName,string valueName,Object
value)
```

说明

设置指定的注册表项的指定名称/值对。如果指定的项不存在，则创建该项。

SetValue 方法的各参数说明如表 S.15 所示。

表 S.15　SetValue 方法的各参数说明

参　数	描　述
keyName	以有效注册表根（如“HKEY_CURRENT_USER”）开头的键的完整注册表路径
valueName	名称/值对的名称
value	要存储的值

- RegistryKey 对象的 SetValue 方法

```
public void SetValue (string name, Object value)
```


更多资源请访问稀酷客(www.ckook.com)



说明

设置指定的名称/值对。

SetValue 方法的各参数说明如表 S.16 所示。

表 S.16 SetValue 方法的各参数说明

参　数	描　述
name	要存储的值的名称
value	要存储的数据



命名空间

Microsoft.Win32



示例

本示例使用 Registry 对象的 SetValue 方法在 HKEY_CURRENT_CONFIG 键下创建一个名为 mrsoft 的子键，然后在 mrsoft 子键下添加一个名为 mr 的键值项，同时为这个键值项添加键值 hywork。

```
Registry.SetValue("HKEY_CURRENT_CONFIG\\mrsoft", "mr", "hywork");
```

本示例用 RegistryKey 对象的 SetValue 方法在 HKEY_CURRENT_CONFIG 键下创建一个名为 mrsoft 的子键，然后在 mrsoft 子键下添加一个名为 mr 的键值项，同时为这个键值项添加键值 mr。

```
RegistryKey var_reg = Registry.CurrentConfig.CreateSubKey
("HKEY_CURRENT_CONFIG\\mrsoft");
var_reg.SetValue("mr", "mr");
```

Show 方法



语法

- Cursor 类的 Show 方法

```
public static void Show ()
```



说明

显示鼠标。

更多资源请访问稀酷客(www.ckook.com)

- Form 类的 Show 方法

```
public void Show ()
```

说明

用来显示窗体或控件。

ShowDialog 方法

语法

- Form 类的 ShowDialog 方法

```
public DialogResult ShowDialog ()
```

说明

将窗体显示为模式对话框，并将当前活动窗口设置为它的所有者。

- object 类的 ShowDialog 方法

```
object. ShowDialog()
```

说明

用默认的所有者运行通用对话框。

返回值：DialogResult 值之一。DialogResult 值是一个枚举值，DialogResult 值及说明如表 S.17 所示。

表 S.17　DialogResult 值及说明

名　称	描　述
Abort	对话框的返回值是 Abort（通常从标签为“中止”的按钮发送）
Cancel	对话框的返回值是 Cancel（通常从标签为“取消”的按钮发送）
Ignore	对话框的返回值是 Ignore（通常从标签为“忽略”的按钮发送）
No	对话框的返回值是 No（通常从标签为“否”的按钮发送）
None	从对话框返回了 Nothing。这表明有模式对话框继续运行
OK	对话框的返回值是 OK（通常从标签为“确定”的按钮发送）
Retry	对话框的返回值是 Retry（通常从标签为“重试”的按钮发送）
Yes	对话框的返回值是 Yes（通常从标签为“是”的按钮发送）

更多资源请访问稀酷客(www.ckook.com)

ShowToday 属性

获取或设置一个值，该值指示控件底端是否显示 TodayDate 属性表示的日期。

语法

```
public void SetToolTip (Control control,string caption)
```

属性值：如果显示的是今天的日期，为 true，否则为 false。默认为 true。

示例

本示例用 ShowToday 方法在 MonthCalendar 控件显示时，在底部不显示今天日期。

```
monthCalendar1.ShowToday = false;//不显示今天日期
```

Sleep 方法

将当前线程挂起指定的时间。

语法

```
public static void Sleep (int millisecondsTimeout)
```

Sleep 方法的参数说明如表 S.18 所示。

表 S.18　Sleep 方法的参数说明

参　数	描　述
millisecondsTimeout	是指线程被阻止的毫秒数。指定零（0）以指示应挂起此线程以使其他等待线程能够执行。指定 Infinite 以无限期阻止线程

示例

本示例用 Sleep 方法睡眠主线程 30 毫秒。

```
Thread.Sleep(30);//睡眠主线程
```

本示例用 Sleep 方法挂起主线程。

```
Thread.Sleep(0);//挂起主线程
```

SMTP 协议

功能

SMTP 是简单邮件传输协议，它提供客户端向服务器端发送邮件的功能，

更多资源请访问稀酷客(www.ckook.com)

即客户端向服务器发出请求指令，服务器端给出应答。

SMTP 分为命令头和信息头两部分，命令头主要完成客户端与服务器端的连接和验证等功能，整个过程由多条命令组成。由服务器给出响应信息，一般为 3 位数字的响应码和响应文本，不同的服务器回送响应码是遵守该协议的，而响应正文则不必遵守。SMTP 命令及响应信息都是单行，信息体是邮件正文部分，最后应以单独的“.”作为结束行。下面是一些常用的 SMTP 指令：

HELO hostname：与服务器连接并告知客户端使用的机器名，可以随便填写。

MAIL FROMsender_id：告诉服务器发信人的地址。

RCPT TOreceiver_id：告诉服务器收信人地址。

DATA：传输信件内容，且最后要以只含“.”的特殊行结束。

RESET：取消刚才的指令，重新开始。

VERIFY userid：检测账号是否存在（此指令为可选指令，服务器可能不支持）。

QUIT：退出连接。



注意

SMTP 服务器使用的端口号一般为 25，另外，此实例还需要引用 System.Net.Mail 命名空间。



典型应用

本实例通过使用 SMTP 协议，实现发送电子邮件功能。实例运行结果如图 S.5、图 S.6 所示。

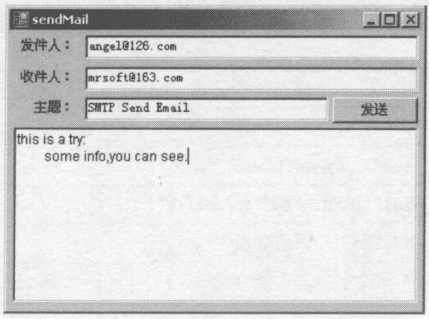


图 S.5 “发送邮件”窗体

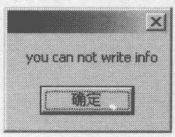


图 S.6 提示信息

更多资源请访问稀酷客(www.ckook.com)

主要代码如下：

添加必要的命名空间，代码如下：

```
using System.Net.Mail;
using System.IO;
using System.Net;
```

自定义一个 CreateTimeoutTestMessage 方法，该方法中通过 MailMessage 类与 SmtpClient 类实例化 Mail 对象与 Client 对象，然后使用 Mail 对象来完成对邮件信息的定制，如收件人、主题和内容等，最后调用 Client 对象的 Send 方法来完成信息的发送功能。CreateTimeoutTestMessage 方法关键代码如下：

```
public  void CreateTimeoutTestMessage(string server)
{
    try
    {
        Mail = new MailMessage(this.txtFrom.Text, this.txtGet.Text);
        Mail.Subject =this.txtSubject.Text.Trim().ToString();
        Mail.Body = this.richTextBox1.Text.Trim().ToString();
        Client = new SmtpClient(server, 25);
        Client.Send(Mail);
        MessageBox.Show("Success! ! ! ");
    }
    catch (Exception ey)
    {
        MessageBox.Show(ey.Message);
    }
}
```

单击“发送”按扭时，程序调用 CreateTimeoutTestMessage 方法完成邮件发送功能。“发送”按扭的 Click 事件代码如下：

```
private void button1_Click(object sender, EventArgs e)
{
    if (this.richTextBox1.Text.Trim().ToString() != "")
    {
        CreateTimeoutTestMessage("hywork");
    }
    else
    {
        MessageBox.Show("you can not write info");
        return;
    }
}
```


更多资源请访问稀酷客(www.ckook.com)



注意

MailMessage 类还提供群发功能以及发送附件（通过 Attachments 类）的功能。

SmtpClient 类



功能

SmtpClient 类用于将电子邮件发送到 SMTP 服务器以便传递。在使用 SmtpClient 类发送电子邮件时，必须指定以下信息：

- 用来发送电子邮件的 SMTP 主机服务器。
- 身份验证凭据（如果 SMTP 服务器要求）。
- 发件人的电子邮件地址。
- 收件人的电子邮件地址。
- 邮件内容。



注意

若要在电子邮件中包括附件，必须先使用 Attachment 类创建附件，然后再使用 MailMessage.Attachments 属性将附件添加到邮件中。



属性

SmtpClient 类的常用属性及说明如表 S.19 所示。

表 S.19　SmtpClient 类的常用属性及说明

属　性	描　述
Credentials	获取用于存储附加到此电子邮件的数据的附件集合
DeliveryMethod	指定如何处理待发的电子邮件
EnableSsl	指定 SmtpClient 对象是否使用安全套接字层（SSL）加密连接
Host	获取或设置用于 SMTP 服务的主机的名称或 IP 地址
Port	获取或设置用于 SMTP 服务的端口
ServicePoint	获取用于传输电子邮件的网络连接

更多资源请访问稀酷客(www.ckook.com)

方法

SmtpClient 类的常用方法及说明如表 S.20 所示。

表 S.20　SmtpClient 类的常用方法及说明

方　法	描　述
Send	将电子邮件发送到 SMTP 服务器以便传递，该方法在传输邮件的过程中将阻止其他操作
SendAsync	发送电子邮件，该方法不会阻止调用线程

Socket 类

功能

这个低层的类用于管理连接，实现 Berkeley 通信端接口，定义绑定、连接网络端点以及传输数据所需的方法，提供处理端点连接传输等细节所需要的功能。WebRequest、TcpClient 和 UdpClinet 等类在内部使用这个类。

命名空间

System.Net.Sockets

属性

Socket 类的常用属性及说明如表 S.21 所示。

表 S.21　Socket 类的常用属性及说明

属　性	描　述
AddressFamily	获取 Socket 的地址族
Available	获取已经从网络接收且可供读取的数据量
Blocking	获取或设置一个值，该值指示 Socket 是否处于阻止模式
Connected	获取一个值，该值指示 Socket 是在上次 Send 还是 Receive 操作时连接到远程主机
DontFragment	获取或设置 Boolean 值，该值指定 Socket 是否允许将 Internet 协议（IP）数据报分段
EnableBroadcast	获取或设置一个 Boolean 值，该值指定 Socket 是否可以发送或接收广播数据包
ExclusiveAddressUse	获取或设置 Boolean 值，该值指定 Socket 是否仅允许一个进程绑定到端口
Handle	获取 Socket 的操作系统句柄
LingerState	获取或设置一个值，该值指定 Socket 在尝试发送所有挂起数据时是否延迟关闭套接字

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
LocalEndPoint	获取本地终结点
MulticastLoopback	获取或设置一个值，该值指定传出的多路广播数据包是否传递到发送应用程序
NoDelay	获取或设置 Boolean 值，该值指定流 Socket 是否正在使用 Nagle 算法
ProtocolType	获取 Socket 的协议类型
ReceiveBufferSize	获取或设置一个值，它指定 Socket 接收缓冲区的大小
ReceiveTimeout	获取或设置一个值，该值指定之后同步 Receive 调用将超时的时间长度
RemoteEndPoint	获取远程终结点
SendBufferSize	获取或设置一个值，该值指定 Socket 发送缓冲区的大小
SendTimeout	获取或设置一个值，该值指定之后同步 Send 调用将超时的时间长度
SocketType	获取 Socket 的类型

方法

Socket 类的常用方法及说明如表 S.22 所示。

表 S.22　Socket 类的常用方法及说明

方　法	描　述
BeginConnect	已重载。开始一个对远程主机连接的异步请求
BeginSend	已重载。将数据异步发送到连接的 Socket
BeginSendTo	向特定远程主机异步发送数据
BeginSendFile	已重载。将文件异步发送到连接的 Socket 对象
Listen	将 Socket 置于侦听状态
SendTo	已重载。将数据发送到特定终结点
Shutdown	禁用某 Socket 上的发送和接收
EndSend	已重载。结束挂起的异步发送
EndSendFile	结束文件的挂起异步发送
EndSendTo	结束挂起的、向指定位置进行的异步发送

SolidBrush 类

功能

定义单色画笔。画笔用于填充图形形状，如矩形、椭圆、扇形、多边形和封闭路径。

更多资源请访问稀酷客(www.ckook.com)

属性

SolidBrush 类的常用属性及说明如表 S.23 所示。

表 S.23　SolidBrush 类的常用属性及说明

属　性	描　述
Color	获取或设置此 SolidBrush 对象的颜色

方法

SolidBrush 类的常用方法及说明如表 S.24 所示。

表 S.24　SolidBrush 类的常用方法及说明

方　法	描　述
Clone	已重写。 创建此 SolidBrush 对象的一个精确副本
Dispose	释放由此 Brush 对象使用的所有资源（从 Brush 继承）

Sort 方法

语法

- Array 类中的 Sort 方法

```
public static void Sort (Array array)
public static void Sort (Array keys,Array items)
public static void Sort (Array keys,Array items)
public static void Sort (Array array,IComparer comparer)
```

说明

使用 Array 中每个元素的 IComparable 实现，对整个一维 Array 中的元素进行排序。

Sort 方法的参数说明如表 S.25 所示。

表 S.25　Sort 方法的参数说明

参　数	描　述
array	要排序的一维 Array
keys	一维 Array，它包含要排序的关键字
items	一维 Array，它包含与 keysArray 中的每一个关键字对应的项
comparer	比较元素时要使用的 IComparer 实现

更多资源请访问稀酷客(www.ckook.com)

- DataGridView 控件中的 Sort 方法

```
public virtual void Sort(DataGridViewColumn dataGridViewColumn,
ListSortDirection direction)
```

说明

基于指定列的内容，按升序或降序对 DataGridView 控件的内容进行排序。

Sort 方法的参数说明如表 S.26 所示。

表 S.26 Sort 方法的参数说明

参　数	描　述
dataGridViewColumn	对 DataGridView 的内容进行排序时所依据的列
direction	ListSortDirection 值之一

ListSortDirection 值的成员说明如表 S.27 所示。

表 S.27 ListSortDirection 值的成员说明

成　员	描　述
ArgumentException	（1）指定的列不属于此 DataGridView （2）DataSource 属性已设置，而指定列的 IsDataBound 属性返回 false
ArgumentNullException	dataGridViewColumn 为空引用
InvalidOperationException	（1）VirtualMode 属性设置为 true，而指定列的 IsDataBound 属性返回 false （2）DataSource 属性指定的对象未实现 IBindingList 接口 （3）DataSource 属性指定的对象的 IBindingList.SupportsSorting 属性值为 false

Split 方法

语法

- Regex 类的 Split 方法

```
String[] Split (string input)
String[] Split (string input,int count)
String[] Split (string input,int count,int startat)
Static String[] Split(string input,string pattern)
```


更多资源请访问稀酷客(www.ckook.com)



说明

在由正则表达式匹配项定义的位置将输入字符串拆分为一个子字符串数组。

Split 方法的参数说明如表 S.28 所示。

表 S.28 Split 方法的参数说明

参　数	描　述
input	要拆分的字符串
count	要返回的最大数组元素数
startat	输入字符串中开始搜索的字符位置
pattern	要匹配的正则表达式

- String 类的 Split 方法

```
    public string[] Split (params char[] separator)
    public string[] Split (char[] separator,int count)
    public string[] Split (char[] separator,StringSplitOptions options)
    public string[] Split (string[] separator,StringSplitOptions options)
    public string[] Split (char[] separator,int
count,StringSplitOptions options)
```



说明

返回包含此实例中的子字符串（由指定 Char 或 String 数组的元素分隔）的 String 数组。

Split 方法的参数说明如表 S.29 所示。

表 S.29 Split 方法的参数说明

参　数	描　述
separator	分隔此实例中子字符串的 Unicode 字符数组、不包含分隔符的空数组或空引用
count	要返回的子字符串的最大数量
options	指定 RemoveEmptyEntries 以省略返回的数组中的空数组元素，或指定 None 以包含返回的数组中的空数组元素

SQL 注入式攻击



功能

SQL 注入式攻击是指利用设计上的漏洞，在目标服务器上运行 Sql 命令

更多资源请访问稀酷客(www.ckook.com)

以及进行其他方式的攻击，动态生成 Sql 命令时没有对用户输入的数据进行验证，此为 Sql 注入攻击得逞的主要原因。

示例

如果用户的查询语句是：

```
select * from tbUser where name='"&user&"' and password='"&pwd&"'",
```

那么，如果用户名为：1' or '1'='1，则查询语句将会变成：

```
select * from admin where tbUser='1 or '1'='1' and password='"&pwd&"'"
```

SqlCommand 类

功能

表示要对 SQL Server 数据库执行的一个 Transact-SQL 语句或存储过程。

命名空间

System.Data.SqlClient

属性

SqlCommand 类的常用属性及说明如表 S.30 所示。

表 S.30　SqlCommand 类的常用属性及说明

属　性	描　述
CommandText	已重写。获取或设置要对数据源执行的 Transact-SQL 语句或存储过程
CommandTimeout	已重写。获取或设置在终止执行命令的尝试并生成错误之前的等待时间
CommandType	已重写。获取或设置一个值，该值指示如何解释 CommandText 属性
Connection	获取或设置 SqlCommand 的此实例使用的 SqlConnection
Container	获取 IContainer，它包含 Component（从 Component 继承）
DesignTimeVisible	已重写。获取或设置一个值，该值指示命令对象是否应在 Windows 窗体设计器控件中可见
Notification	获取或设置一个指定与此命令绑定的 SqlNotificationRequest 对象的值
Site	获取或设置 Component 的 ISite（从 Component 继承）
Transaction	获取或设置将在其中执行 SqlCommand 的 SqlTransaction
UpdatedRowSource	已重写。获取或设置命令结果在由 DbDataAdapter 的“Update”方法使用时，如何应用于 DataRow

更多资源请访问稀酷客(www.ckook.com)

方法

SqlCommand 类的常用方法及说明如表 S.31 所示。

表 S.31　SqlCommand 类的常用方法及说明

方　法	描　述
Cancel	已重写。尝试取消 SqlCommand 的执行
Clone	创建作为当前实例副本的新 SqlCommand 对象
CreateParameter	创建 SqlParameter 对象的新实例
Dispose	已重载。释放由 Component 占用的资源
EndExecuteNonQuery	完成 Transact-SQL 语句的异步执行
EndExecuteReader	完成 Transact-SQL 语句的异步执行，返回请求的 SqlDataReader
EndExecuteXmlReader	完成 Transact-SQL 语句的异步执行，将请求的数据以 XML 形式返回
ExecuteNonQuery	已重写。对连接执行 Transact-SQL 语句并返回受影响的行数
ExecuteReader	已重载。将 CommandText 发送到 Connection 并生成一个 SqlDataReader
ExecuteScalar	已重写。执行查询，并返回查询所返回的结果集中第一行的第一列。忽略其他列或行
ExecuteXmlReader	将 CommandText 发送到 Connection 并生成一个 XmlReader 对象
ResetCommandTimeout	将 CommandTimeout 属性重置为其默认值

事件

SqlCommand 类的常用事件及说明如表 S.32 所示。

表 S.32　SqlCommand 类的常用事件及说明

事　件	描　述
Disposed	添加事件处理程序以侦听组件上的 Disposed 事件
StatementCompleted	当 Transact-SQL 语句执行完成时发生

SqlConnection 类

功能

表示 SQL Server 数据库的一个打开的连接。

命名空间

System.Data.SqlClient

属性

SqlConnection 类的常用属性及说明如表 S.33 所示。

更多资源请访问稀酷客(www.ckook.com)

表 S.33　SqlConnection 类的常用属性及说明

属　性	描　述
ConnectionString	已重写。获取或设置用于打开 SQL Server 数据库的字符串
ConnectionTimeout	已重写。获取在尝试建立连接时终止尝试并生成错误之前所等待的时间
Database	已重写。获取当前数据库或连接打开后要使用的数据库的名称
DataSource	已重写。获取要连接的 SQL Server 实例的名称
PacketSize	获取用来与 SQL Server 的实例通信的网络数据包的大小
Site	获取或设置 Component 的 ISite
State	已重写。指示 SqlConnection 的状态
StatisticsEnabled	如果设置为 true，则对当前连接启用统计信息收集
WorkstationId	获取标识数据库客户端的一个字符串



方法

SqlConnection 类的常用方法及说明如表 S.34 所示。

表 S.34　SqlConnection 类的常用方法及说明

方　法	描　述
BeginTransaction	已重载。开始数据库事务
ChangeDatabase	已重写。为打开的 SqlConnection 更改当前数据库
ChangePassword	将连接字符串中指示的用户的 SQL Server 密码更改为提供的新密码
ClearAllPools	清空连接池
ClearPool	清空与指定连接关联的连接池
Close	已重写。关闭与数据库的连接。这是关闭任何打开连接的首选方法
CreateCommand	创建并返回一个与 SqlConnection 关联的 SqlCommand 对象
CreateObjRef	创建一个对象，该对象包含生成用于与远程对象进行通信的代理所需的全部相关信息
Dispose	已重载。释放由 Component 占用的资源
Open	已重写。使用 ConnectionString 所指定的属性设置打开数据库连接



事件

SqlConnection 类的常用事件及说明如表 S.35 所示。

表 S.35　SqlConnection 类的常用事件及说明

事　件	描　述
InfoMessage	当 SQL Server 返回一个警告或信息性消息时发生
StateChange	当事件状态更改时发生

更多资源请访问稀酷客(www.ckook.com)

SqlDataAdapter 类

功能

表示用于填充 DataSet 和更新 SQL Server 数据库的一组数据命令和一个数据库连接。

命名空间

System.Data.SqlClient

属性

SqlDataAdapter 类的常用属性及说明如表 S.36 所示。

表 S.36 SqlDataAdapter 类的常用属性及说明

属　性	描　述
AcceptChangesDuringFill	获取或设置一个值，该值指示在任何 Fill 操作过程中，在将 AcceptChanges 添加到 DataTable 之后是否在 DataRow 上调用它
AcceptChangesDuringUpdate	获取或设置在 Update 期间是否调用 AcceptChanges
DeleteCommand	获取或设置一个 Transact-SQL 语句或存储过程，以从数据集删除记录
FillLoadOption	获取或设置 LoadOption，后者确定适配器如何从 DbDataReader 中填充 DataTable
InsertCommand	获取或设置一个 Transact-SQL 语句或存储过程，以在数据源中插入新记录
MissingMappingAction	确定传入数据没有匹配的表或列时需要执行的操作
MissingSchemaAction	确定现有 DataSet 架构与传入数据不匹配时需要执行的操作
SelectCommand	获取或设置一个 Transact-SQL 语句或存储过程，用于在数据源中选择记录
TableMappings	获取一个集合，它提供源表和 DataTable 之间的主映射
UpdateBatchSize	获取或设置每次到服务器的往返过程中处理的行数
UpdateCommand	获取或设置一个 Transact-SQL 语句或存储过程，用于更新数据源中的记录

方法

SqlDataAdapter 类的常用方法及说明如表 S.37 所示。

更多资源请访问稀酷客(www.ckook.com)

表 S.37　SqlDataAdapter 类的常用方法及说明

方　法	描　述
Dispose	已重载。释放由 Component 占用的资源
Fill	已重载。填充 DataSet 或 DataTable
FillSchema	已重载。将 DataTable 添加到 DataSet 中，并配置架构以匹配数据源中的架构
GetFillParameters	获取当执行 SQL SELECT 语句时由用户设置的参数
ResetFillLoadOption	将 FillLoadOption 重置为默认状态，并使 Fill 接受 AcceptChangesDuringFill
Update	已重载。为 DataSet 中每个已插入、已更新或已删除的行调用相应的 INSERT、UPDATE 或 DELETE 语句

事件

SqlDataAdapter 类的常用事件及说明如表 S.38 所示。

表 S.38　SqlDataAdapter 类的常用事件及说明

事　件	描　述
Disposed	添加事件处理程序以侦听组件上的 Disposed 事件
FillError	在填充操作过程中出现错误时返回
RowUpdated	在对数据源执行命令后的 Update 过程中发生。试图进行更新，因此激发了该事件
RowUpdating	在对数据源执行命令前的 Update 过程中发生。试图进行更新，因此激发了该事件

SqlDataReader 类

功能

提供一种从 SQL Server 数据库读取行的只进流的方式。

命名空间

System.Data.SqlClient

属性

SqlDataReader 类的常用属性及说明如表 S.39 所示。

更多资源请访问稀酷客(www.ckook.com)

表 S.39 SqlDataReader 类的常用属性及说明

属　性	描　述
Depth	已重写。获取一个值，用于指示当前行的嵌套深度
FieldCount	已重写。获取当前行中的列数
HasRows	已重写。获取一个值，该值指示 SqlDataReader 是否包含一行或多行
IsClosed	已重写。检索一个布尔值，该值指示是否已关闭指定的 SqlDataReader 实例
Item	已重载。已重写。获取以本机格式表示的列的值
RecordsAffected	已重写。获取执行 Transact-SQL 语句所更改、插入或删除的行数
VisibleFieldCount	已重写。获取 SqlDataReader 中未隐藏的字段的数目



方法

SqlDataReader 类的常用方法及说明如表 S.40 所示。

表 S.40 SqlDataReader 类的常用方法及说明

方　法	描　述
Close	已重写。关闭 SqlDataReader 对象
Dispose	已重载。释放由 DbDataReader 占用的资源
GetByte	已重写。获取指定列的字节形式的值
GetChar	已重写。获取指定列的单个字符串形式的值
GetData	返回被请求的列序号的 DbDataReader 对象
GetDataTypeName	已重写。获取源数据类型的名称
GetDateTime	已重写。获取指定列的 DateTime 对象形式的值
GetDecimal	已重写。获取指定列的 Decimal 对象形式的值
GetDouble	已重写。获取指定列的双精度浮点数形式的值
GetEnumerator	已重写。返回循环访问 SqlDataReader 的 IEnumerator
GetFieldType	已重写。获取是对象的数据类型的 Type
GetFloat	已重写。获取指定列的单精度浮点数形式的值
GetGuid	已重写。获取指定列的值作为全局唯一标识符（GUID）
GetInt16	已重写。获取指定列的 16 位有符号整数形式的值
GetInt32	已重写。获取指定列的 32 位有符号整数形式的值
GetInt64	已重写。获取指定列的 64 位有符号整数形式的值
GetName	已重写。获取指定列的名称
GetOrdinal	已重写。在给定列名称的情况下获取列序号
GetSchemaTable	已重写。返回一个 DataTable，它描述 SqlDataReader 的列元数据
GetSqlBinary	获取指定列的 SqlBinary 形式的值
GetSqlBoolean	获取指定列的 SqlBoolean 形式的值

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
GetSqlByte	获取指定列的 SqlByte 形式的值
GetSqlBytes	获取指定列的 SqlBytes 形式的值
GetSqlChars	获取指定列的 SqlChars 形式的值
GetSqlDateTime	获取指定列的 SqlDateTime 形式的值
GetSqlDecimal	获取指定列的 SqlDecimal 形式的值
GetSqlDouble	获取指定列的 SqlDouble 形式的值
GetSqlGuid	获取指定列的 SqlGuid 形式的值
GetSqlInt16	获取指定列的 SqlInt16 形式的值
GetSqlInt32	获取指定列的 SqlInt32 形式的值
GetSqlInt64	获取指定列的 SqlInt64 形式的值
GetSqlMoney	获取指定列的 SqlMoney 形式的值
GetSqlSingle	获取指定列的 SqlSingle 形式的值
GetSqlString	获取指定列的 SqlString 形式的值
GetSqlValues	获取当前行中的所有属性列
GetString	已重写。获取指定列的字符串形式的值
GetValue	已重写。获取以本机格式表示的指定列的值
GetValues	已重写。获取当前行的集合中的所有属性列
IsDBNull	已重写。获取一个值，该值指示列中是否包含不存在的或已丢失的值
NextResult	已重写。当读取批处理 Transact-SQL 语句的结果时，使数据读取器前进到下一个结果
Read	已重写。使 SqlDataReader 前进到下一条记录

Stack 类

功能

表示对象的简单的后进先出非泛型集合。

命名空间

System.Collections

属性

Stack 类的常用属性及说明如表 S.41 所示。

更多资源请访问稀酷客(www.ckook.com)

表 S.41　Stack 类的常用属性及说明

属　性	描　述
Count	获取 Stack 中包含的元素数
IsSynchronized	获取一个值，该值指示是否同步对 Stack 的访问（线程安全）
SyncRoot	获取可用于同步 Stack 访问的对象

方法

Stack 类的常用方法及说明如表 S.42 所示。

表 S.42　Stack 类的常用方法及说明

方　法	描　述
Clear	从 Stack 中移除所有对象
Clone	创建 Stack 的浅表副本
Contains	确定某元素是否在 Stack 中
CopyTo	从指定数组索引开始将 Stack 复制到现有一维 Array 中
Peek	返回位于 Stack 顶部的对象但不将其移除
Pop	移除并返回位于 Stack 顶部的对象
Push	将对象插入 Stack 的顶部
ToArray	将 Stack 复制到新数组中

Start 方法

启动进程资源并将其与 Process 组件关联。

语法

```
    public bool Start()
    public static Process Start (ProcessStartInfo startInfo)
    public static Process Start (string fileName)
    public static Process Start (string fileName,string arguments)
    public static Process Start (string fileName,string
userName,SecureString password,    string domain)
```

返回值：与进程资源关联的新的 Process 组件，或者如果没有启动进程资源，则为空引用。

Start 方法的各参数说明如表 S.43 所示。

表 S.43　Start 方法的各参数说明

参　数	描　述
startInfo	ProcessStartInfo，包含用于启动进程的信息

更多资源请访问稀酷客(www.ckook.com)

续表

参　数	描　述
fileName	要在进程中运行的文档或应用程序文件的名称
arguments	启动该进程时传递的命令行参数
userName	启动进程时使用的用户名
password	一个 SecureString，它包含启动进程时要使用的密码
domain	启动进程时要使用的域

命名空间

System.Diagnostics

StartPosition 属性

获取或设置运行时窗体的起始位置。

语法

```
public FormStartPosition StartPosition { get; set; }
```

属性值：FormStartPosition，表示窗体的起始位置。FormStartPosition 枚举的成员说明如表 S.44 所示。

表 S.44　FormStartPosition 枚举的成员说明

成　员	描　述
CenterParent	窗体在其父窗体中居中
CenterScreen	窗体在当前显示窗口中居中，其尺寸在窗体大小中指定
Manual	窗体的位置由 Location 属性确定
WindowsDefaultBounds	窗体定位在 Windows 默认位置，其边界也由 Windows 默认决定
WindowsDefaultLocation	窗体定位在 Windows 默认位置，其尺寸在窗体大小中指定

Step 属性

获取或设置调用 PerformStep 方法增加进度栏的当前位置时所根据的数量。

语法

```
public int Step { get; set; }
```

属性值：每次调用 PerformStep 方法增加进度栏时所根据的数量。默认值为 10。

更多资源请访问稀酷客(www.ckook.com)

Stop 方法

语法

```
public void Stop ()
```

说明

（1）在 WebBrowser 类中此方法用于取消所有挂起的导航并停止所有动态页元素（如背景声音和动画）。

（2）在 ServiceController 类中此方法用于停止该服务以及任何依赖于该服务的服务。

Stream 类

功能

提供字节序列的一般视图。Stream 类是读写字节数据的数据流类中最基础的基类，这一组类中包含了 Stream 类及其下面派生的几个类，这些派生的类包括内存数据流（MemoryStream）、文件数据流（FileStream）以及缓冲数据流（BufferedStream），全部派生自 Stream 类，专门用来处理字节数据的读写以及文件的随机访问等相关操作。

命名空间

System.IO

属性

Stream 类的常用属性及说明如表 S.45 所示。

表 S.45　Stream 类的常用属性及说明

属　性	描　述
CanRead	当在派生类中重写时，获取指示当前流是否支持读取的值
CanSeek	当在派生类中重写时，获取指示当前流是否支持查找功能的值
CanTimeout	获取一个值，该值确定当前流是否可以超时
CanWrite	当在派生类中重写时，获取指示当前流是否支持写入功能的值
Length	当在派生类中重写时，获取用字节表示的流长度
Position	当在派生类中重写时，获取或设置当前流中的位置

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
ReadTimeout	获取或设置一个值，该值确定流在超时前尝试读取多长时间
WriteTimeout	获取或设置一个值，该值确定流在超时前尝试写入多长时间

方法

Stream 类的常用方法及说明如表 S.46 所示。

表 S.46　Stream 类的常用方法及说明

方　法	描　述
BeginRead	开始异步读操作
BeginWrite	开始异步写操作
Close	关闭当前流并释放与之关联的所有资源（如套接字和文件句柄）
Dispose	已重载。释放由 Stream 对象使用的所有资源
EndRead	等待挂起的异步读取完成
EndWrite	结束异步写操作
Equals	已重载。确定两个 Object 实例是否相等
Flush	当在派生类中重写时，将清除该流的所有缓冲区，并使得所有缓冲数据被写入到基础设备
Read	当在派生类中重写时，从当前流读取字节序列，并将此流中的位置提升读取的字节数
ReadByte	从流中读取一个字节，并将流内的位置向前推进一个字节，或者如果已到达流的末尾，则返回-1
Seek	当在派生类中重写时，设置当前流中的位置
SetLength	当在派生类中重写时，设置当前流的长度
Write	当在派生类中重写时，向当前流中写入字节序列，并将此流中的当前位置提升写入的字节数
WriteByte	将一个字节写入流内的当前位置，并将流内的位置向前推进一个字节

StreamReader 类

功能

实现一个 TextReader，使其以一种特定的编码从字节流中读取字符。

命名空间

System.IO

更多资源请访问稀酷客(www.ckook.com)

属性

StreamReader 类的常用属性及说明如表 S.47 所示。

表 S.47 StreamReader 类的常用属性及说明

属　性	描　述
BaseStream	返回基础流
CurrentEncoding	获取当前 StreamReader 对象正在使用的当前字符编码
EndOfStream	获取一个值，该值表示当前的流位置是否在流的末尾

方法

StreamReader 类的常用方法及说明如表 S.48 所示。

表 S.48 StreamReader 类的常用方法及说明

方　法	描　述
Close	已重写。关闭 StringReader
Dispose	释放由此 TextReader 对象使用的所有资源
Peek	返回下一个可用的字符，但不使用它
Read	读取输入字符串中的下一个字符或下一组字符
ReadBlock	从当前流中读取最大 count 的字符并从 index 开始将该数据写入 buffer
ReadLine	从基础字符串中读取一行
ReadToEnd	将整个流或从流的当前位置到流的结尾作为字符串读取

StreamWriter 类

功能

实现一个 TextWriter，使其以一种特定的编码向流中写入字符。

命名空间

System.IO

属性

StreamWriter 类的常用属性及说明如表 S.49 所示。

表 S.49 StreamWriter 类的常用属性及说明

属　性	描　述
AutoFlush	获取或设置一个值，该值指示 StreamWriter 是否在每次调用 StreamWriter.Write 之后，将其缓冲区刷新到基础流

更多资源请访问稀酷客(www.ckook.com)

属　性	描　述
BaseStream	获取同后备存储区连接的基础流
Encoding	已重写。获取将输出写入到其中的 Encoding
FormatProvider	获取控制格式设置的对象
NewLine	获取或设置由当前 TextWriter 使用的行结束符字符串

方法

StreamWriter 类的常用方法及说明如表 S.50 所示。

表 S.50　StreamWriter 类的常用方法及说明

方　法	描　述
Close	已重写。关闭当前的 StreamWriter 对象和基础流
Dispose	已重载。释放由此 TextWriter 对象使用的所有资源
Flush	已重写。清理当前编写器的所有缓冲区，并使所有缓冲数据写入基础流
Write	已重载。写入流
WriteLine	已重载。写入重载参数指定的某些数据，后跟行结束符

String 类

功能

表示文本，即一系列 Unicode 字符。

属性

String 类的常用属性及说明如表 S.51 所示。

表 S.51　String 类的常用属性及说明

属　性	描　述
Chars	获取此实例中位于指定字符位置的字符
Length	获取此实例中的字符数

方法

String 类的常用方法及说明如表 S.52 所示。

更多资源请访问稀酷客(www.ckook.com)

表 S.52 String 类的常用方法及说明

方 法	描 述
Clone	返回对此 String 实例的引用
Compare	已重载。比较两个指定的 String 对象
CompareOrdinal	已重载。通过计算每个字符串中相应 Char 对象的数值来比较两个 String 对象
CompareTo	已重载。将此实例与指定的对象或 String 进行比较，并返回二者相对值的指示
Concat	已重载。 连接 String 的一个或多个实例，或 Object 的一个或多个实例的值的 String 表示形式
Contains	返回一个值，该值指示指定的 String 对象是否出现在此字符串中
Copy	创建一个与指定的 String 具有相同值的 String 的新实例
CopyTo	将指定数目的字符从此实例中的指定位置复制到 Unicode 字符数组中的指定位置
EndsWith	已重载。确定 String 的实例的末尾是否与指定的字符串匹配
Format	已重载。 将指定的 String 中的每个格式项替换为相应对象的值的文本等效项
IndexOf	已重载。报告 String 或一个或多个字符在此字符串中的第一个匹配项的索引
IndexOfAny	已重载。报告指定 Unicode 字符数组中的任意字符在此实例中第一个匹配项的索引
Insert	在此实例中的指定索引位置插入一个指定的 String 实例
Intern	检索系统对指定 String 的引用
IsInterned	检索对指定 String 的引用
IsNormalized	已重载。指示此字符串是否符合特定的 Unicode 范式
IsNullOrEmpty	指示指定的 String 对象是空引用，还是 Empty 字符串
Join	已重载。在指定 String 数组的每个元素之间串联指定的分隔符 String，从而产生单个串联的字符串
LastIndexOf	已重载。报告指定的 Unicode 字符或 String 在此实例中的最后一个匹配项的索引位置
LastIndexOfAny	已重载。报告在 Unicode 数组中指定的一个或多个字符在此实例中的最后一个匹配项的索引位置
Normalize	已重载。返回一个新字符串，其二进制表示形式符合特定的 Unicode 范式
op_Equality	确定两个指定的 String 对象是否具有同一值
op_Inequality	确定两个指定的 String 对象是否具有不同的值
PadLeft	已重载。右对齐此实例中的字符，在左边用空格或指定的 Unicode 字符填充以达到指定的总长度
PadRight	已重载。左对齐此字符串中的字符，在右边用空格或指定的 Unicode 字符填充以达到指定的总长度

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
Remove	已重载。从此实例中删除指定个数的字符
Replace	已重载。将此实例中的指定 Unicode 字符或 String 的所有匹配项替换为其他指定的 Unicode 字符或 String
Split	已重载。返回包含此实例中的子字符串（由指定 Char 或 String 数组的元素分隔）的 String 数组
StartsWith	已重载。确定 String 实例的开头是否与指定的字符串匹配
Substring	已重载。从此实例检索子字符串
ToCharArray	已重载。将此实例中的字符复制到 Unicode 字符数组
ToLower	已重载。返回此 String 转换为小写形式的副本
ToLowerInvariant	返回此 String 对象的转换为小写形式的副本，返回时使用固定区域性的大小写规则
ToString	已重载。已重写。将此实例的值转换为 String
ToUpper	已重载。返回此 String 转换为大写形式的副本
ToUpperInvariant	返回此 String 对象的转换为大写形式的副本，返回时使用固定区域性的大小写规则
Trim	已重载。从此实例的开始位置和末尾移除一组指定字符的所有匹配项
TrimEnd	从此实例的结尾移除数组中指定的一组字符的所有匹配项
TrimStart	从此实例的开始位置移除数组中指定的一组字符的所有匹配项

StringBuilder 类

功能

表示可变字符字符串。

属性

StringBuilder 类的常用属性及说明如表 S.53 所示。

表 S.53　StringBuilder 类的常用属性及说明

属　性	描　述
Capacity	获取或设置可包含在当前实例所分配的内存中的最大字符数
Chars	获取或设置此实例中指定字符位置处的字符
Length	获取或设置此实例的长度
MaxCapacity	获取此实例的最大容量

更多资源请访问稀酷客(www.ckook.com)

方法

StringBuilder 类的常用方法及说明如表 S.54 所示。

表 S.54　StringBuilder 类的常用方法及说明

方　法	描　述
Append	已重载。在此实例的结尾追加指定对象的字符串表示形式
AppendFormat	已重载。在此实例追加包含零个或更多格式规范的格式化字符串。每个格式规范由相应对象参数的字符串表示形式替换
AppendLine	已重载。将默认的行终止符（或指定字符串的副本和默认的行终止符）追加到此实例的末尾
CopyTo	将此实例的指定段中的字符复制到目标 Char 数组的指定段中
EnsureCapacity	确保 StringBuilder 的此实例的容量至少是指定值
Insert	已重载。将指定对象的字符串表示形式插入到此实例中的指定字符位置
Remove	将指定范围的字符从此实例中移除
Replace	已重载。将此实例中所有的指定字符或字符串替换为其他的指定字符或字符串

Substring 方法

从指定字符串中截取子串。

语法

```
public string Substring (int startIndex,int length)
```

Substring 方法的参数说明如表 S.55 所示。

表 S.55　Substring 方法的参数说明

参　数	描　述
startIndex	子字符串的起始位置的索引
length	子字符串中的字符数

SUM 函数

返回表达式中所有值的和，或只返回 DISTINCT 值。SUM 只能用于数字列。空值将被忽略。

语法

```
SUM ( [ ALL | DISTINCT ] expression )
```


更多资源请访问稀酷客(www.ckook.com)

返回值：以最精确的 expression 数据类型返回所有表达式值的和。

SUM 函数的参数说明如表 S.56 所示。

表 S.56　SUM 函数的参数说明

参　数	描　述
ALL	对所有的值进行聚合函数运算。ALL 是默认设置
DISTINCT	指定 SUM 返回唯一值的和
Expression	是常量、列或函数，或者是算术、按位与字符串等运算符的任意组合。expression 是精确数字或近似数字数据类型分类（bit 数据类型除外）的表达式。不允许使用聚合函数和子查询

switch case 语句

switch 语句是一个控制语句，它通过将控制传递给其体内的一个 case 语句来处理多个选择和枚举。switch 语句中有很多 case 区段，每一个 case 标记后可以指定一个常数作为标准，不能将一组常数放在一个 case 标记之后。

语法

```
switch(判断条件)
{
case 判断式:
      //区段，符合条件执行的代码
break;
......
[default:]
}
```

switch 语句可以包括任意数目的 case 实例，但是任何两个 case 语句都不能具有相同的值。语句体从选定的语句开始执行，直到 break 将控制传递到 case 体以外。在每一个 case 块的后面，都必须有一个跳转语句（如 break）。但有一个例外，（与 C++中的 switch 语句不同）C#不支持从一个 case 标签显式贯穿到另一个 case 标签，这个例外是当 case 语句中没有代码时。如果没有任何 case 表达式与开关值匹配，则控制传递给跟在可选 default 标签后的语句。如果没有 default 标签，则控制传递到 switch 以外。

更多资源请访问稀酷客(www.ckook.com)

T

TabControl 控件

TabControl 控件显示多个选项卡，这些选项卡类似于笔记本中的分隔卡和档案柜文件夹中的标签。TabControl 控件的选项卡中可包含图片和其他控件。此外，TabControl 控件还可以用来创建用于设置一组相关属性的属性页。

属性

TabControl 控件的常用属性及说明如表 T.1 所示。

表 T.1 TabControl 控件的常用属性及说明

属 性	描 述
ItemSize	此属性获取或设置控件的选项卡的大小
Multiline	此属性获取或设置一个值，该值指示是否可以显示一行以上的选项卡
RowCount	此属性获取控件的选项卡条中当前正显示的行数
SelectedIndex	此属性获取或设置当前选定的选项卡页的索引
SelectedTab	此属性获取或设置当前选定的选项卡页
ShowToolTips	此属性获取或设置一个值，该值指示当鼠标移到选项卡上时是否显示该选项卡的“工具提示”
TabPages	此属性获取该选项卡控件中选项卡页的集合
Size	获取或设置控件的高度和宽度
SizeMode	获取或设置调整控件的选项卡大小的方式
TabCount	获取选项卡条中选项卡的数目
ClientSize	获取或设置控件的工作区的高度和宽度
Alignment	获取或设置选项卡在其中对齐的控件区域
CanFocus	获取一个值，该值指示控件是否可以接收焦点
ImageList	获取或设置在控件的选项卡上显示的图像

方法

TabControl 控件的常用方法及说明如表 T.2 所示。

表 T.2 TabControl 控件的常用方法及说明

方 法	描 述
Show	向用户显示控件
SuspendLayout	临时挂起控件的布局逻辑

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
Select	激活控件
Hide	对用户隐藏控件
FindForm	检索控件所在的窗体
Contains	检索一个值，该值指示指定控件是否为一个控件的子控件
SelectTab	已重载。使指定的选项卡成为当前选项卡

事件

TabControl 控件的常用事件及说明如表 T.3 所示。

表 T.3　TabControl 控件的常用事件及说明

事　件	描　述
Click	在单击控件时发生
Deselected	当取消选择选项卡时发生
Deselecting	在取消选择某个选项卡之前发生，使处理程序能够取消选项卡更改
DoubleClick	在双击控件时发生
Move	在移动控件时发生
Selected	当选择某个选项卡时发生

TabIndex 属性

获取或设置在控件的容器的控件的 Tab 键顺序。

语法

```
[LocalizableAttribute(true)]
public int TabIndex { get; set; }
```

属性值：控件容器中控件集合内的控件的索引值。容器中包括的控件以 Tab 键顺序排列。

注意

Tab 索引可由任何大于等于零的有效整数组成，越小的数字在 Tab 键顺序中越靠前。如果在同一父控件上多个控件具有相同的 Tab 索引，则控件的 Z 顺序确定控件的循环顺序。为了按 Tab 键顺序包括控件，其 TabStop 属性必须设置为 True。

更多资源请访问稀酷客(www.ckook.com)

Tables 属性

获取包含在 DataSet 中的表的集合。

语法

```
public DataTableCollection Tables { get; }
```

属性值：此 DataSet 对象包含的 DataTableCollection 对象。如果 DataTable 对象不存在，将返回空集合。



注意

若要向集合中添加表，请使用 DataTableCollection 的 Add 方法。若要移除表，请使用 Remove 方法。

TabPages 属性

获取该选项卡控件中选项卡页的集合。

语法

```
public TabPageCollection TabPages { get; }
```

属性值：TabControl.TabPageCollection，它包含 TabControl 中的 TabPage 对象。

示例

本示例是用 TabPages 属性获取第一个选项卡的名称。

```
string V_Name = tabControl1.TabPages[0].Text;
```

本示例是用 TabPages 属性在第一个选项卡上显示 imageList1 组件中的第二个图标。

```
tabControl1.ImageList = imageList1;
tabControl1.TabPages[0].ImageIndex = 1;
```

本示例是用 TabPages 属性判断当前选中的选项卡是否在集合中，如果在则显示当前选项卡的名称。

```
    private void tabControl1_Selected(object sender, TabControl
EventArgs e)
    {
```


更多资源请访问稀酷客(www.ckook.com)

```
    if (((TabControl)sender).TabPages.Contains(((TabControl)sender).
SelectedTab))//判断当前选项卡是否在集合中
    {
        //显示当前选项卡的名称
        MessageBox.Show(((TabControl)sender).SelectedTab.Text);
    }
}
```

Tag 属性

获取或设置一个对象，该对象包含控件的数据。

语法

```
[LocalizableAttribute(false)]
[BindableAttribute(true)]
public Object Tag { get; set; }
```

属性值：一个 Object，它包含有关控件的数据。默认为空引用。

TcpClient 类

功能

为 TCP 网络服务提供客户端连接。

命名空间

System.Net.Sockets

属性

TcpClient 类的常用属性及说明如表 T.4 所示。

表 T.4　TcpClient 类的常用属性及说明

属　性	描　述
Available	获取已经从网络接收且可供读取的数据量
Client	获取或设置基础 Socket
Connected	获取一个值，该值指示 TcpClient 的基础 Socket 是否已连接到远程主机
ExclusiveAddressUse	获取或设置 Boolean 值，该值指定 TcpClient 是否只允许一个客户端使用端口

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
LingerState	获取或设置有关套接字逗留时间的信息
NoDelay	获取或设置一个值，该值在发送或接收缓冲区未满时禁用延迟
ReceiveBufferSize	获取或设置接收缓冲区的大小
ReceiveTimeout	获取或设置在初始化一个读取操作以后 TcpClient 等待接收数据的时间量
SendBufferSize	获取或设置发送缓冲区的大小
SendTimeout	获取或设置 TcpClient 等待发送操作成功完成的时间量

方法

TcpClient 类的常用方法及说明如表 T.5 所示。

表 T.5　TcpClient 类的常用方法及说明

方　法	描　述
BeginConnect	开始一个对远程主机连接的异步请求
Close	释放此 TcpClient 实例，而不关闭基础连接
Connect	使用指定的主机名和端口号将客户端连接到 TCP 主机
EndConnect	异步接受传入的连接尝试
GetStream	返回用于发送和接收数据的 NetworkStream

TcpListener 类

功能

从 TCP 网络客户端侦听连接。

命名空间

System.Net.Sockets

属性

TcpListener 类的常用属性及说明如表 T.6 所示。

表 T.6　TcpListener 类的常用属性及说明

属　性	描　述
Active	获取一个值，该值指示 TcpListener 是否正主动侦听客户端连接

更多资源请访问稀酷客(www.ckook.com)

方法

TcpListener 类的常用方法及说明如表 T.7 所示。

表 T.7 TcpListener 类的常用方法及说明

方 法	描 述
AcceptSocket	接受挂起的连接请求
AcceptTcpClient	接受挂起的连接请求
BeginAcceptSocket	开始一个异步操作来接受一个传入的连接尝试
BeginAcceptTcpClient	开始一个异步操作来接受一个传入的连接尝试
EndAcceptSocket	异步接受传入的连接尝试，并创建新的 Socket 来处理远程主机通信
EndAcceptTcpClient	异步接受传入的连接尝试，并创建新的 TcpClient 来处理远程主机通信
Pending	确定是否有挂起的连接请求
Start	开始侦听传入的连接请求
Stop	关闭侦听器

Text 属性

用来获取或设置与控件关联的文本。

语法

```
[BindableAttribute(true)]
[LocalizableAttribute(true)]
public override string Text { get; set; }
```

属性值：与该控件关联的文本。

TextBox 控件

TextBox 控件用于获取用户输入或显示文本，它通常用于可编辑文本，有时也可使其成为只读控件。另外，TextBox 控件还可以显示密码或多行。

属性

TextBox 控件的常用属性及说明如表 T.8 所示。

表 T.8 TextBox 控件的常用属性及说明

属 性	描 述
AcceptsReturn	用于指示在多行编辑控件中是否允许输入回车符
Mmaxlength	指定可以编辑控件中输入的最大字符数

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
Mulitiline	是否可以在文本框中输入多行文本
PasswordChar	表示所显示的字符而不是实际输入的文本，当接受用户输入的机密数据和敏感数据时，通常需要设置此属性
ReadOnly	确定文本框中文本是否是只读
TextAlign	指示文本框中文本以何种方式对齐
Scrollbars	指定是否在多行 TextBox 控件中显示滚动条
SelectedText	获取或设置一个值，该值指示控件中当前选定的文本
SelectionLength	获取或设置文本框中选定的字符数
SelectionStart	获取或设置文本框中选定的文本起始点

方法

TextBox 控件的常用方法及说明如表 T.9 所示。

表 T.9　TextBox 控件的常用方法及说明

方　法	描　述
AppendText	用于大文本框中当前显示的文本末尾添加指定的文本
Clear	用于清空文本框中当前显示的文本
Copy	用于将文本框中的选定文本复制到剪贴板
Cut	删除文本框中的选定文本并将其放置在剪贴板
Paste	用于将剪贴板中文本替换文本框中选定的文本
Select	已重载。选择控件中的文本
SelectAll	选定文本框中的所有文本

事件

TextBox 控件的常用事件及说明如表 T.10 所示。

表 T.10　TextBox 控件的常用事件及说明

事　件	描　述
TextChanged	在 Text 属性值更改时发生
Click	在单击控件时发生
DockChanged	当 Dock 属性的值更改时发生
Enter	进入控件时发
MultilineChanged	在 Multiline 属性的值更改后发生

TextChanged 事件

在 Text 属性值更改时发生。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public event EventHandler TextChanged
```

TextReader 类

功能

表示可读取连续字符系列的读取器。

命名空间

System.IO

方法

TextReader 类的常用方法及说明如表 T.11 所示。

表 T.11　TextReader 类的常用方法及说明

方　法	描　述
Close	关闭 TextReader 并释放与该 TextReader 关联的所有系统资源
Dispose	已重载。释放由此 TextReader 对象使用的所有资源
Peek	读取下一个字符，而不更改读取器状态或字符源。返回下一个可用字符，而实际上并不从输入流中读取此字符
Read	已重载。从输入流读取数据
ReadBlock	从当前流中读取最大 count 的字符并从 index 开始将该数据写入 buffer
ReadLine	从当前流中读取一行字符并将数据作为字符串返回
ReadToEnd	读取从当前位置到 TextReader 的结尾的所有字符并将它们作为一个字符串返回

TextWriter 类

功能

表示可以编写一个有序字符系列的编写器。该类为抽象类。

命名空间

System.IO

更多资源请访问稀酷客(www.ckook.com)

属性

TextWriter 类的常用属性及说明如表 T.12 所示。

表 T.12　TextWriter 类的常用属性及说明

属　性	描　述
Encoding	当在派生类中重写时，返回用来写输出的 Encoding
FormatProvider	获取控制格式设置的对象
NewLine	获取或设置由当前 TextWriter 使用的行结束符字符串

方法

TextWriter 类的常用方法及说明如表 T.13 所示。

表 T.13　TextWriter 类的常用方法及说明

方　法	描　述
Close	关闭当前编写器并释放任何与该编写器关联的系统资源
Dispose	已重载。释放由此 TextWriter 对象使用的所有资源
Flush	清理当前编写器的所有缓冲区，使所有缓冲数据写入基础设备
Write	已重载。将给定数据类型写入文本流
WriteLine	已重载。写入重载参数指定的某些数据，后跟行结束符

Thread 类

功能

创建并控制线程，设置其优先级并获取其状态。

命名空间

System.Threading

属性

Thread 类的常用属性及说明如表 T.14 所示。

表 T.14　Thread 类的常用属性及说明

属　性	描　述
ApartmentState	获取或设置此线程的单元状态
CurrentContext	获取线程正在其中执行的当前上下文
CurrentCulture	获取或设置当前线程的区域性

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
CurrentPrincipal	获取或设置线程的当前负责人
CurrentThread	获取当前正在运行的线程
CurrentUICulture	获取或设置资源管理器使用的当前区域性以便在运行时查找区域性特定的资源
ExecutionContext	获取一个 ExecutionContext 对象，该对象包含有关当前线程的各种上下文的信息
IsAlive	获取一个值，该值指示当前线程的执行状态
IsBackground	获取或设置一个值，该值指示某个线程是否为后台线程
IsThreadPoolThread	获取一个值，该值指示线程是否属于托管线程池
ManagedThreadId	获取当前托管线程的唯一标识符
Name	获取或设置线程的名称
Priority	获取或设置一个值，该值指示线程的调度优先级
ThreadState	获取一个值，该值包含当前线程的状态



方法

Thread 类的常用方法及说明如表 T.15 所示。

表 T.15　Thread 类的常用方法及说明

方　法	描　述
Abort	已重载。在调用此方法的线程上引发 ThreadAbortException，以开始终止此线程的过程。调用此方法通常会终止线程
AllocateDataSlot	在所有的线程上分配未命名的数据槽
AllocateNamedDataSlot	在所有线程上分配已命名的数据槽
BeginCriticalRegion	通知宿主执行将要进入一个代码区域，在该代码区域内线程中止或未处理的异常的影响可能会危害应用程序域中的其他任务
BeginThreadAffinity	通知宿主托管代码将要执行依赖于当前物理操作系统线程的标识的指令
EndCriticalRegion	通知宿主执行将要进入一个代码区域，在该代码区域内线程中止或未处理的异常仅影响当前任务
EndThreadAffinity	通知宿主托管代码已执行完依赖于当前物理操作系统线程的标识的指令
FreeNamedDataSlot	为进程中的所有线程消除名称与槽之间的关联
GetApartmentState	返回一个 ApartmentState 值，该值指示单元状态
GetCompressedStack	返回一个 CompressedStack 对象，该对象可用于捕获当前线程的堆栈
GetData	在当前线程的当前域中从当前线程上指定的槽中检索值

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
GetDomain	返回当前线程正在其中运行的当前域
GetDomainID	返回唯一的应用程序域标识符
GetHashCode	已重写。返回当前线程的哈希代码
GetNamedDataSlot	查找已命名的数据槽
Interrupt	中断处于 WaitSleepJoin 线程状态的线程
Join	已重载。阻止调用线程，直到某个线程终止时为止
MemoryBarrier	同步内存。其效果是将缓存内存中的内容刷新到主内存中，从而使处理器能执行当前线程
ResetAbort	取消为当前线程请求的 Abort
Resume	继续已挂起的线程
SetApartmentState	在线程启动前设置其单元状态
SetCompressedStack	对当前线程应用捕获的 CompressedStack
SetData	在当前正在运行的线程上为此线程的当前域在指定槽中设置数据
Sleep	已重载。将当前线程阻止指定的毫秒数
SpinWait	导致线程等待由 iterations 参数定义的时间量
Start	已重载。使线程被安排进行执行
Suspend	挂起线程，或者如果线程已挂起，则不起作用
TrySetApartmentState	在线程启动前设置其单元状态
VolatileRead	已重载。读取字段值。无论处理器的数目或处理器缓存的状态如何，该值都是由计算机的任何处理器写入的最新值
VolatileWrite	已重载。立即向字段写入一个值，以使该值对计算机中的所有处理器都可见

ThreadState 属性

获取一个值，该值包含当前线程的状态。



语法

```
public ThreadState ThreadState { get; }
```

属性值：ThreadState 值之一，它指示当前线程的状态。初始值为 Unstarted。ThreadState 枚举的成员及说明如表 T.16 所示。

表 T.16　ThreadState 枚举的成员及说明

成　员	描　述
Initialized	此状态指示线程已初始化但尚未启动
Ready	此状态指示线程因无可用的处理器而等待使用处理器。线程准备在下一个可用的处理器上运行

更多资源请访问稀酷客(www.ckook.com)

续表

成　员	描　述
Running	此状态指示线程当前正在使用处理器
Standby	此状态指示线程将要使用处理器。一次只能有一个线程处于此状态
Terminated	此状态指示线程已完成执行并已退出
Transition	此状态指示线程在可以执行前等待处理器之外的资源。例如，它可能正在等待其执行堆栈从磁盘中分页
Unknown	线程的状态未知
Wait	此状态指示线程尚未准备好使用处理器，因为它正在等待外围操作完成或等待资源释放。当线程就绪后，将对其进行重排

命名空间

System.Threading

throw 语句

功能

throw 语句用于发出在程序执行期间出现异常情况的信号。throw 语句通常与 try-catch 或 try-finally 语句一起使用。可以使用 throw 语句显式引发异常（这里引发自定义异常）。创建用户自定义异常，好的编码方法是以“Exception”作为用户自定义异常类名的结尾。

语法

```
throw  ExObject
```

Tick 事件

该事件在当指定的计时器间隔已过去而且计时器处于启用状态时发生。

语法

```
public event EventHandler Tick
```

Timer 组件

定期引发事件的组件，该组件是为 Windows 窗体环境设计的。

更多资源请访问稀酷客(www.ckook.com)

属性

Timer 组件的常用属性及说明如表 T.17 所示。

表 T.17 Timer 组件的常用属性及说明

属　性	描　述
Enabled	获取或设置计时器是否正在运行
Interval	获取或设置计时器开始计时之间的时间（以毫秒为单位）
Site	获取或设置在设计模式中将 Timer 绑定到其容器的站点

方法

Timer 组件的常用方法及说明如表 T.18 所示。

表 T.18 Timer 组件的常用方法及说明

方　法	描　述
Start	启动计时器
Stop	停止计时器

事件

Timer 组件的常用事件及说明如表 T.19 所示。

表 T.19 Timer 组件的常用事件及说明

事　件	描　述
Tick	该事件在当指定的计时器间隔已过去而且计时器处于启用状态时发生

TimeSpan 对象

功能

表示时间间隔或持续时间，按正负天数、小时数、分钟数、秒数以及秒的小数部分进行度量。

属性

TimeSpan 对象的常用属性及说明如表 T.20 所示。

表 T.20 TimeSpan 对象的常用属性及说明

属　性	描　述
Days	获取由当前 TimeSpan 结构表示的整天数
Hours	获取由当前 TimeSpan 结构表示的整小时数

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
Milliseconds	获取由当前 TimeSpan 结构表示的整毫秒数
Minutes	获取由当前 TimeSpan 结构表示的整分钟数
Seconds	获取由当前 TimeSpan 结构表示的整秒数
Ticks	获取表示当前 TimeSpan 结构的值的刻度数
TotalDays	获取以整天数和天的小数部分表示的当前 TimeSpan 结构的值
TotalHours	获取以整小时数和小时的小数部分表示的当前 TimeSpan 结构的值
TotalMilliseconds	获取以整毫秒数和毫秒的小数部分表示的当前 TimeSpan 结构的值
TotalMinutes	获取以整分钟数和分钟的小数部分表示的当前 TimeSpan 结构的值
TotalSeconds	获取以整秒数和秒的小数部分表示的当前 TimeSpan 结构的值



方法

TimeSpan 对象的常用方法及说明如表 T.21 所示。

表 T.21　TimeSpan 对象的常用方法及说明

方　法	描　述
Add	将指定的 TimeSpan 添加到此实例中
Compare	比较两个 TimeSpan 值并返回一个指示它们之间关系的整数
CompareTo	已重载。将此实例与指定的对象或 TimeSpan 进行比较，并返回二者相对值的指示
Duration	返回新的 TimeSpan 对象，其值是当前 TimeSpan 对象的绝对值
Equals	已重载。已重写。返回一个值，该值指示 TimeSpan 的两个实例是否相等
FromDays	返回表示指定天数的 TimeSpan，其中对天数的指定精确到最接近的毫秒
FromHours	返回表示指定小时数的 TimeSpan，其中对小时数的指定精确到最接近的毫秒
FromMilliseconds	返回表示指定毫秒数的 TimeSpan
FromMinutes	返回表示指定分钟数的 TimeSpan，其中对分钟数的指定精确到最接近的毫秒
FromSeconds	返回表示指定秒数的 TimeSpan，其中对秒数的指定精确到最接近的毫秒
FromTicks	返回表示指定时间的 TimeSpan，其中对时间的指定以刻度为单位
Negate	返回其值为此实例的相反值的 TimeSpan
Subtract	从此实例中减去指定的 TimeSpan
ToString	已重写。返回此实例的值的字符串表示形式

更多资源请访问稀酷客(www.ckook.com)

ToInt32 方法

返回由字节数组中指定位置的四个字节转换来的 32 位有符号整数。

语法

```
public static uint ToUInt32 (string value,int fromBase)
```

返回值：等效于 value 中的数字的 32 位有符号整数。如果 value 为空，则返回值为零。

ToInt32 方法的各参数及说明如表 T.22 所示。

表 T.22　ToInt32 方法的各参数及说明

参　数	描　述
value	包含数字的 String
fromBase	value 中数字的基数，必须是 2、8、10 或 16

ToLower 方法

语法

- Char 类的 ToLower 方法

```
public static char ToLower(char c)
public static char ToLower(char c,CultureInfo culture)
```

说明

使用指定的区域性特定格式设置信息将指定 Unicode 字符的值转换为它的小写等效项。

返回值：c 的小写等效项；如果 c 已经是小写形式或不是字母，则为 c 的未更改值。

ToLower 方法的各参数及说明如表 T.23 所示。

表 T.23　ToLower 方法的各参数及说明

参　数	描　述
c	一个 Unicode 字符
culture	提供区域性特定的大小写规则的 System.Globalization.CultureInfo 对象，或者为空引用

更多资源请访问稀酷客(www.ckook.com)

- String 类的 ToLower 方法

```
public string ToLower()
public string ToLower(CultureInfo culture)
```

说明

返回此 String 转换为小写形式的副本。

返回值：小写形式的 String。

ToLower 方法的各参数及说明如表 T.24 所示。

表 T.24　ToLower 方法的各参数及说明

参　数	描　述
culture	CultureInfo 对象，它提供区域性特定的大小写规则

ToolStrip 控件

ToolStrip 控件是可以在 Windows 窗体应用程序中承载菜单、控件和用户控件的工具条。

属性

ToolStrip 控件的常用属性及说明如表 T.25 所示。

表 T.25　ToolStrip 控件的常用属性及说明

属　性	描　述
AutoSize	已重写。获取或设置一个值，该值指示是否自动调整控件的大小以完整显示其内容
BackgroundImage	获取或设置在控件中显示的背景图像
BackgroundImageLayout	获取或设置 ImageLayout 枚举中定义的背景图像布局
BindingContext	已重写。获取或设置 ToolStrip 的绑定上下文
CanOverflow	获取或设置一个值，该值指示是否可以将 ToolStrip 中的项发送到溢出菜单
CanSelect	获取一个值，该值指示是否可以选中控件
GripStyle	获取或设置 ToolStrip 移动手柄是可见还是隐藏
ImageList	获取或设置包含 ToolStrip 项上显示的图像的图像列表
ImageScalingSize	获取或设置 ToolStrip 上所用图像的大小
Items	获取属于 ToolStrip 的所有项
LayoutStyle	获取或设置一个值，该值指示 ToolStrip 如何对项集合进行布局

更多资源请访问稀酷客(www.ckook.com)

方法

ToolStrip 控件的常用方法及说明如表 T.26 所示。

表 T.26　ToolStrip 控件的常用方法及说明

方　法	描　述
Contains	检索一个值，该值指示指定控件是否为一个控件的子控件
FindForm	检索控件所在的窗体
Focus	为控件设置输入焦点
GetChildAtPoint	已重载。检索指定位置的子控件
GetContainerControl	沿着控件的父控件链向上，返回下一个 ContainerControl
Hide	对用户隐藏控件
ScrollControlIntoView	将指定的子控件滚动到支持自动滚动的控件的视图中
SelectNextControl	激活下一个控件
Show	向用户显示控件

事件

ToolStrip 控件的常用事件及说明如表 T.27 所示。

表 T.27　ToolStrip 控件的常用事件及说明

事　件	描　述
Click	在单击控件时发生
ItemAdded	当向 ToolStripItemCollection 添加新的 ToolStripItem 时发生
ItemClicked	在单击 ToolStripItem 时发生
TextChanged	在 Text 属性值更改时发生
VisibleChanged	在 Visible 属性值更改时发生

ToolTip 控件

该控件是一个长方形的弹出窗口，在用户将指针悬停在一个控件上时，该窗口显示有关该控件用途的简短说明。

属性

ToolTip 控件的常用属性及说明如表 T.28 所示。

表 T.28　ToolTip 控件的常用属性及说明

属　性	描　述
Active	此属性获取或设置一个值，指示工具提示当前是否活动
AutoPopDelay	此属性获取或设置当指针在具有指定工具提示文本的控件内保持静止时，工具提示保持可见的时间期限（以毫秒为单位）

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
InitialDelay	此属性获取或设置工具提示显示之前经过的时间，才会显示 ToolTip 控件
ShowAlways	此属性获取或设置一个值，该值指示是否显示工具提示窗口，甚至是在其父控件不活动的时候
UseAnimation	此属性获取或设置一个值，该值确定在显示工具提示时是否应使用动画效果



方法

ToolTip 控件的常用方法及说明如表 T.29 所示。

表 T.29　ToolTip 控件的常用方法及说明

方　法	描　述
Dispose	此方法释放由工具提示占用的资源
GetToolTip	此方法检索与指定控件关联的工具提示文本
RemoveAll	此方法移除当前与工具提示组件关联的所有工具提示文本
SetToolTip	此方法使工具提示文本与指定的控件相关联
Show	此方法设置与工具提示关联的文本，然后显示该工具提示

ToolTipIcon 属性

获取或设置一个值，该值定义要在工具提示文本旁显示的图标的类型。



语法

```
public ToolTipIcon ToolTipIcon { get; set; }
```

属性值：System.Windows.Forms.ToolTipIcon 枚举值之一。ToolTipIcon 枚举的成员及说明如表 T.30 所示。

表 T.30　ToolTipIcon 枚举的成员及说明

成　员	描　述
Error	错误图标
Info	信息图标
None	不是标准图标
Warning	警告图标

ToolTipText 属性

用来获取或设置作为控件的 ToolTip 显示的文本。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public string ToolTipText { get; set; }
```

属性值：一个表示工具提示文本的字符串。

ToolTipTitle 属性

获取或设置工具提示窗口的标题。

语法

```
public string ToolTipTitle { get; set; }
```

属性值：包含窗口标题的 String。

本示例用 ToolTipText 属性设置提示文本框的标题。

```
private void textBox1_MouseEnter(object sender, EventArgs e)
{
    toolTip1.ToolTipIcon = ToolTipIcon.Warning;
    toolTip1.ToolTipTitle = "提示";//设置提示文本框的标题
    toolTip1.SetToolTip(textBox1, "只能输入数值型数据");
}
```

运行结果如图 T.1 所示。



图 T.1　ToolTipText 属性的应用

Top 属性

获取或设置控件上边缘与其容器的工作区上边缘之间的距离（以像素为单位）。

语法

```
public int Top { get; set; }
```

属性值：Int32 表示控件下边缘与其容器的工作区上边缘之间的距离（以像素为单位）。

更多资源请访问稀酷客(www.ckook.com)

示例

本示例在 textBox1 控件中设置 button1 按钮的高度，并通过 Top 属性来进行修改。

```
private void button1_Click(object sender, EventArgs e)
{
    button1.Top = Convert.ToInt32(textBox1.Text);
}
```

TopMost 属性

获取或设置一个值，该值指示窗体是否应显示为应用程序的最顶层窗体。

语法

```
public bool TopMost { get; set; }
```

属性值：如果将窗体显示为最顶层窗体，则为 True；否则为 False。默认为 False。

ToShortDateString 方法

此方法将此实例的值转换为其等效的短日期字符串表示形式。

语法

```
public string ToShortDateString ()
```

返回值：一个字符串，它包含与此实例的日期值等效的数字月份、该月中的数字日期和年份。

示例

本示例用 ToShortDateString 方法将系统日期转换为短日期格式（当前的日期和时间为：2008-11-3 13:40:21）。

```
DateTime.Now.ToShortDateString();
```

运行结果为：2008-11-3

ToShortTimeString 方法

此方法将此实例的值转换为其等效的短时间字符串表示形式。

更多资源请访问稀酷客(www.ckook.com)

语法

```
public string ToShortTimeString ()
```

返回值：与此实例中的时间值等效的一个字符串，其中包含当日的周日名称、当月的名称以及小时、分和秒的数字日期。

ToString 方法

返回表示当前 Object 的 String。

语法

```
public virtual string ToString()
public static string ToString(bool b)
public static string ToString(double d)
```

返回值：String 类型，表示当前的 Object。

ToString 方法的各参数及说明如表 T.31 所示。

表 T.31　ToString 方法的各参数及说明

参　数	描　述
b	布尔型值
d	双精度值

ToUpper 方法

语法

- Char 类的 ToUpper 方法

```
public static char ToUpper(char c)
public static char ToUpper(char c,CultureInfo culture)
```

说明

使用指定的区域性特定格式设置信息将指定 Unicode 字符的值转换为它的大写等效项。

返回值：c 的大写等效项；如果 c 已经是大写形式或不是字母，则为 c 的未更改值。

更多资源请访问稀酷客(www.ckook.com)

ToUpper 方法的各参数及说明如表 T.32 所示。

表 T.32　ToUpper 方法的各参数及说明

参　数	描　述
c	一个 Unicode 字符
culture	提供区域性特定的大小写规则的 System.Globalization.CultureInfo 对象，或者为空引用（在 Visual Basic 中为 Nothing）

- String 类的 ToUpper 方法

```
public string ToUpper()
public string ToUpper(CultureInfo culture)
```



说明

返回此 String 转换为大写形式的副本。

返回值：大写形式的 String。

ToUpper 方法的各参数及说明如表 T.33 所示。

表 T.33　ToUpper 方法的各参数及说明

参　数	描　述
culture	CultureInfo 对象，它提供区域性特定的大小写规则

TrackBar 控件

表示一个标准的 Windows 跟踪条。



属性

TrackBar 控件的常用属性及说明如表 T.34 所示。

表 T.34　TrackBar 控件的常用属性及说明

属　性	描　述
BackColor	获取或设置控件的背景色
BackgroundImage	已重写。获取或设置 TrackBar 控件的背景图像
BackgroundImageLayout	已重写。获取或设置一个 ImageLayout 值；但设置此属性对 TrackBar 控件没有影响
ContextMenu	获取或设置与控件关联的快捷菜单
Focused	获取一个值，该值指示控件是否有输入焦点
ForeColor	已重写。获取跟踪条的前景色

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
LargeChange	获取或设置一个值，当滚动框长距离移动时向 Value 属性加上该值或从中减去该值
Maximum	获取或设置此 TrackBar 使用的范围的上限
MaximumSize	获取或设置大小，该大小是 GetPreferredSize 可以指定的上限
Minimum	获取或设置此 TrackBar 使用的范围的下限
MinimumSize	获取或设置大小，该大小是 GetPreferredSize 可以指定的下限
Orientation	获取或设置一个值，该值指示跟踪条在水平方向还在垂直方向
Value	获取或设置表示跟踪条上滚动框的当前位置的数值
SmallChange	获取或设置当滚动框短距离移动时对 Value 属性进行增减的值
TickFrequency	获取或设置一个值，该值指定控件上绘制的刻度之间的增量
TickStyle	获取或设置一个值，该值指示如何显示跟踪条上的刻度线

方法

TrackBar 控件的常用方法及说明如表 T.35 所示。

表 T.35　TrackBar 控件的常用方法及说明

方　法	描　述
BeginInit	开始初始化在窗体上使用或由另一个组件使用的 TrackBar。初始化发生在运行时
BeginInvoke	已重载。在创建控件的基础句柄所在线程上异步执行委托
Contains	检索一个值，该值指示指定控件是否为一个控件的子控件
Dispose	已重载。释放由 Control 使用的所有资源
FindForm	检索控件所在的窗体
Focus	为控件设置输入焦点
GetTopLevel	确定控件是否是顶级控件
Select	已重载。激活控件

事件

TrackBar 控件的常用事件及说明如表 T.36 所示。

表 T.36　TrackBar 控件的常用事件及说明

事　件	描　述
Click	当用户单击 TrackBar 控件时发生
DoubleClick	当用户双击 TrackBar 控件时发生
DragDrop	在完成拖放操作时发生
DragEnter	在将对象拖入控件的边界时发生

更多资源请访问稀酷客(www.ckook.com)

续表

事　件	描　述
DragLeave	在将对象拖出控件的边界时发生
Enter	进入控件时发生
KeyDown	在控件有焦点的情况下按下键时发生
KeyPress	在控件有焦点的情况下按下键时发生
KeyUp	在控件有焦点的情况下释放键时发生
Leave	在输入焦点离开控件时发生
Scroll	在通过鼠标或键盘操作移动滚动框时发生

TreeNode 类

功能

表示 TreeView 控件中的节点。

属性

TreeNode 类的常用属性及说明如表 T.37 所示。

表 T.37　TreeNode 类的常用属性及说明

属　性	描　述
Checked	获取或设置一个值，该值指示节点的复选框是否被选中
ChildNodes	获取 TreeNodeCollection 集合，该集合包含当前节点的第一级子节点
DataBound	获取一个值，该值指示节点是否是通过数据绑定创建的
DataItem	获取绑定到控件的数据项
DataPath	获取绑定到节点的数据的路径
Depth	获取节点的深度
Expanded	获取或设置一个值，该值指示是否展开节点
ImageToolTip	获取或设置在节点旁边显示的图像的工具提示文本
ImageUrl	获取或设置节点旁显示的图像的 URL
NavigateUrl	获取或设置单击节点时导航到的 URL
Parent	获取当前节点的父节点
PopulateOnDemand	获取或设置一个值，该值指示是否动态填充节点
SelectAction	获取或设置选择节点时引发的事件
Selected	获取或设置一个值，该值指示是否选择节点
ShowCheckBox	获取或设置一个值，该值指示是否在节点旁显示一个复选框
Target	获取或设置用来显示与节点关联的网页内容的目标窗口或框架

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
Text	获取或设置为 TreeView 控件中的节点显示的文本
ToolTip	获取或设置节点的工具提示文本
Value	获取或设置用于存储有关节点的任何其他数据（如用于处理回发事件的数据）的非显示值
ValuePath	获取从根节点到当前节点的路径

方法

TreeNode 类的常用方法及说明如表 T.38 所示。

表 T.38　TreeNode 类的常用方法及说明

方　法	描　述
Collapse	折叠当前树节点
CollapseAll	折叠当前节点及其所有子节点
Expand	展开当前树节点
ExpandAll	展开当前节点及其所有子节点
Select	选择 TreeView 控件中的当前节点
ToggleExpandState	切换节点的展开和折叠状态

TreeView 控件

TreeView 控件可以为用户显示节点层次结构，就像在 Windows 操作系统的资源管理器的左窗格中显示文件和文件夹一样。树视图中的各个节点都可以包含其他节点。

属性

TreeView 控件的常用属性及说明如表 T.39 所示。

表 T.39　TreeView 控件的常用属性及说明

属　性	描　述
ImageIndex	此属性显示节点图像的索引
ImageList	表示可显示节点处的图像列表
Nodes	TreeView 控件中的所有节点
SelectedNode	此属性表示 TreeView 控件中当前选中的节点
Showlines	此属性指定树视图的同级节点之间以及树节点和根节点之间是否有线
ShowPlusminus	指示是否父节点旁边显示加减按钮

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
PathSeparator	获取或设置树节点路径所使用的分隔符串
Scrollable	获取或设置一个值，用以指示树视图控件是否在需要时显示滚动条
ShowRootLines	获取或设置一个值，用以指示是否在树视图根处的树节点之间绘制连线
TopNode	获取树视图控件中第一个完全可见的树节点
LineColor	获取或设置连接 TreeView 控件的节点的线条的颜色

方法

TreeView 控件的常用方法及说明如表 T.40 所示。

表 T.40　TreeView 控件的常用方法及说明

方　法	描　述
CollapseAll	此方法折叠所有树节点。当树节点折叠时，只显示父节点
ExpandAll	此方法展开所有树节点，包括子节点
GetNodeCount	此方法用于获取 Treeview 控件中的节点总数
GetNodeAt	已重载。检索位于指定位置的树节点

事件

TreeView 控件的常用事件及说明如表 T.41 所示。

表 T.41　TreeView 控件的常用事件及说明

事　件	描　述
AfterCheck	选中显示在树节点旁边的复选框，会触发此事件
AfterCollapse	当树节点折叠时，会触发此事件
AfterExpand	当树节点展开时，会触发此事件
AfterSelect	当选中树节点时，会触发此事件
SelectedNodeChanged	当选择 TreeView 控件中的节点时发生
TreeNodeCheckChanged	当 TreeView 控件中的复选框在向服务器的两次发送过程之间状态有所更改时发生
TreeNodeCollapsed	当折叠 TreeView 控件中的节点时发生
TreeNodeExpanded	当扩展 TreeView 控件中的节点时发生

try…catch…finally 语句

异常发生时，执行将终止，同时将控制权交给最近的异常处理语句，这就意味着不能按正常的设计思想执行。而有些资源清理（如关闭文件）必须

更多资源请访问稀酷客(www.ckook.com)

总是执行，即使有异常发生，为了实现该功能，可以使用 finally 块。finally 块用于清除 try 块中分配的任何资源，以及运行必须执行的代码，即使在发生异常的情况下，控制将总是传递给 finally 块，与 try 块的退出方式无关。



语法

```
try
{
    被监控的代码
}
catch(异常类名  异常变量名)
{
    异常处理
}
…
finally
{
    程序代码
}
```

typeof 运算符



功能

typeof 运算符用于获得系统原型对象的类型，也就是 Type 对象。Type 类包含关于值类型和引用类型的信息。typeof 运算符可以在 C#语言中各种位置使用，以找出关于引用类型和值类型的信息。

更多资源请访问稀酷客(www.ckook.com)

UdpClient 类

功能

提供用户数据报（UDP）网络服务。

命名空间

System.Net.Sockets

属性

UdpClient 类的常用属性及说明如表 U.1 所示。

表 U.1　UdpClient 类的常用属性及说明

属　性	描　述
Available	获取从网络接收的可读取的数据量
Client	获取或设置基础网络 Socket
DontFragment	获取或设置 Boolean 值，指定 UdpClient 是否允许对 Internet 协议（IP）数据报进行分段
EnableBroadcast	获取或设置 Boolean 值，指定 UdpClient 是否可以发送或接收广播数据包
ExclusiveAddressUse	获取或设置 Boolean 值，指定 UdpClient 是否只允许一个客户端使用端口
MulticastLoopback	获取或设置 Boolean 值，指定是否将输出多路广播数据包传递给发送应用程序
Ttl	获取或设置一个值，指定由 UdpClient 发送的 Internet 协议（IP）数据包的生存时间（TTL）

方法

UdpClient 类的常用方法及说明如表 U.2 所示。

表 U.2　UdpClient 类的常用方法及说明

方　法	描　述
BeginReceive	从远程主机异步接收数据报
BeginSend	已重载。将数据报异步发送到远程主机

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
Close	关闭 UDP 连接
Connect	已重载。建立默认远程主机
DropMulticastGroup	已重载。退出多路广播组
EndReceive	结束挂起的异步接收
EndSend	结束挂起的异步发送
JoinMulticastGroup	已重载。将 UdpClient 添加到多路广播组
Receive	返回已由远程主机发送的 UDP 数据报
Send	已重载。将 UDP 数据报发送到远程主机

UpButton 方法

将 NumericUpDown 控件的值按照 Increment 属性指定数值递增。

语法

```
public override void UpButton ()
```

Update 方法

使控件重绘其工作区内的无效区域。

语法

```
public void Update()
```

注意

有两种重新绘制窗体及其内容的方法：

（1）可以将 Invalidate 方法的重载之一与 Update 方法一起使用。

（2）可以调用 Refresh 方法，此方法强制控件重新绘制其自身及其所有子级。这等效于将 Invalidate 方法设置为 True 并将该方法与 Update 一起使用。

示例

本示例在移动窗体时，用 Update 方法重绘其工作区内的无效区域。

更多资源请访问稀酷客(www.ckook.com)

```
private void Form1_LocationChanged(object sender, EventArgs e)
{
    this.Update();
}
```

UPDATE 语句

更改表中的现有数据。

语法

```
UPDATE
      {
       table_name WITH ( < table_hint_limited > [ ...n ] )
       | view_name
       | rowset_function_limited
      }
      SET
      { column_name = { expression | DEFAULT | NULL }
      | @variable = expression
      | @variable = column = expression } [ ,...n ]
   { { [ FROM { < table_source > } [ ,...n ] ]
      [ WHERE
         < search_condition > ] }
      |
      [ WHERE CURRENT OF
      { { [ GLOBAL ] cursor_name } | cursor_variable_name }
      ] }
      [ OPTION ( < query_hint > [ ,...n ] ) ]
```

UPDATE 语句的各参数及说明如表 U.3 所示。

表 U.3 UPDATE 语句的各参数及说明

参　数	描　述
table_name	需要更新的表的名称。如果该表不在当前服务器或数据库中，或不为当前用户所有，这个名称可用链接服务器、数据库和所有者名称来限定
WITH (< table_hint _limited > [...n])	指定目标表所允许的一个或多个表提示。需要有 WITH 关键字和圆括号。不允许有 READPAST、NOLOCK 和 READUNCOMMITTED
view_name	要更新的视图的名称。通过 view_name 来引用的视图必须是可更新的。用 UPDATE 语句进行的修改，至多只能影响视图的 FROM 子句所引用的基表中的一个
rowset_function_limited	OPENQUERY 或 OPENROWSET 函数，视提供程序功能而定

更多资源请访问稀酷客(www.ckook.com)

续表

参　数	描　述
SET	指定要更新的列或变量名称的列表
column_name	含有要更改数据的列的名称。column_name 必须驻留于 UPDATE 子句中所指定的表或视图中。标识列不能进行更新

UpdateCommand 属性

获取或设置用于更新数据源中的记录的命令。

语法

```
public DbCommand UpdateCommand { get; set; }
```

属性值：在 Update 过程中使用的 IDbCommand，它针对数据集中已修改的行更新数据源中的记录。

命名空间

System.Data.SqlClient

Url 属性

获取或设置所引用的服务说明的 URL。

语法

```
public override string Url { get; set; }
```

说明

设置此属性等效于向 Navigate 方法传递指定的 URL 并调用该方法。WebBrowser 控件维护浏览会话期间访问的所有网页的历史记录列表。设置 Url 属性时，WebBrowser 控件导航到指定的 URL 并将该 URL 添加到历史记录列表的末尾。

属性值：所引用的服务说明的 URL。

更多资源请访问稀酷客(www.ckook.com)

using 关键字

功能

using 关键字有两个主要用途：

（1）作为指令，用于为命名空间创建别名或导入其他命名空间中定义的类型。

（2）作为语句，用于定义一个范围，在此范围的末尾将释放对象。

- using 指令

using 指令有两个用途：

（1）允许在命名空间中使用类型，以便你不必限定在该命名空间中使用的类型。

（2）为命名空间创建别名。

语法

```
using namespace;
using alias = type|namespace;
```

using 指令中各参数及说明如表 U.4 所示。

表 U.4　using 指令中各参数及说明

参　数	描　述
Alias	希望用来表示命名空间或类型的用户定义符号。可以使用 alias 来表示命名空间名称
Type	alias 表示的类型
namespace	通过 alias 表示的命名空间。或者是一个命名空间，它包含在无须指定完全限定名的情况下使用的类型

更多资源请访问稀酷客(www.ckook.com)

V

Value 属性

获取或设置辅助性对象的值。



语法

```
public virtual string Value { get; set; }
```

属性值：辅助性对象的值，或者，如果此对象没有设置值，则是空引用。

View 属性

此属性用于获取或设置项在控件中的显示方式。



语法

```
public View View { get; set; }
```

属性值：View 值之一，默认为 LargeIcon。View 的属性值及说明如表 V.1 所示。

表 V.1 View 属性的返回值

属　性	返 回 值
Details	每个项显示在不同的行上，并带有关于列中所排列的各项的进一步信息。最左边的列包含一个小图标和标签，后面的列包含应用程序指定的子项。列显示一个标头，它可以显示列的标题。用户可以在运行时调整各列的大小
LargeIcon	每个项都显示为一个最大化图标，在它的下面有一个标签
SmallIcon	每个项都显示为一个小图标，在它的右边有一个标签
List	每个项都显示为一个小图标，在它的右边带一个标签。各项排列在列中，没有列标头



示例

本示例用 View 属性在 ListView 控件的上面添加了一个列标题栏，可以通过该列标题对各列的数据进行说明。

```
private void Form1_Load(object sender, EventArgs e)
{
    listView1.View = View.SmallIcon;
```


更多资源请访问稀酷客(www.ckook.com)

```
    listView1.Items.Clear();//清空所有项的集合
    listView1.Columns.Clear();//清空所有列的集合
    listView1.SmallImageList = imageList1;
    listView1.GridLines = true;//在各数据之间形成网格线
    listView1.View = View.Details;//显示列名称
    listView1.FullRowSelect = true;//在单击某项时，对其进行选中
    //设置头像
    listView1.Columns.Add("头像", 40, HorizontalAlignment.Right);
    //设置头像
    listView1.Columns.Add("用户名", 110, HorizontalAlignment.Center);
    ListViewItem item = new ListViewItem("", 0);
    item.SubItems.Add("张某");
    listView1.Items.Add(item);
    ListViewItem item1 = new ListViewItem("", 1);
    item1.SubItems.Add("孙某");
    listView1.Items.Add(item1);
}
```

运行结果如图 V.1 所示。



图 V.1　View 属性的应用

Visible 属性

用于设置控件是否可见。

语法

```
public bool Visible { get; set; }
```

属性值：如果显示该控件，则为 True，否则为 False。默认为 True。

VscrollBar 控件

表示标准的 Windows 垂直滚动条。

更多资源请访问稀酷客(www.ckook.com)

属性

VscrollBar 控件的常用属性及说明如表 V.2 所示。

表 V.2 VscrollBar 控件的常用属性及说明

属　性	描　述
AutoScrollOffset	获取或设置一个值，该值指示在 ScrollControlIntoView 中将控件滚动到何处
AutoSize	获取或设置一个值，该值指示是否自动调整 ScrollBar 的大小以完整显示其内容
Bottom	获取控件下边缘与其容器的工作区上边缘之间的距离（以像素为单位）
CanFocus	获取一个值，该值指示控件是否可以接收焦点
CanSelect	获取一个值，该值指示是否可以选中控件
Maximum	获取或设置可滚动范围的上限值
MaximumSize	获取或设置大小，该大小是 GetPreferredSize 可以指定的上限
Minimum	获取或设置可滚动范围的值的下限
MinimumSize	获取或设置大小，该大小是 GetPreferredSize 可以指定的下限

方法

VscrollBar 控件的常用方法及说明如表 V.3 所示。

表 V.3 VscrollBar 控件的常用方法及说明

方　法	描　述
Hide	对用户隐藏控件
IsKeyLocked	确定 Caps Lock、Num Lock 或 Scroll Lock 键是否有效
Select	已重载。激活控件
Show	向用户显示控件
Update	使控件重绘其工作区内的无效区域

事件

VscrollBar 控件的常用事件及说明如表 V.4 所示。

表 V.4 VscrollBar 控件的常用事件及说明

事　件	描　述
Click	在单击 ScrollBar 控件时发生
DoubleClick	在双击 ScrollBar 控件时发生
Enter	进入控件时发生
TextChanged	当 Text 属性的值更改时发生

更多资源请访问稀酷客(www.ckook.com)

WebBrowser 控件

使用户可以在窗体中导航网页。WebBrowser 控件为 WebBrowser ActiveX 控件提供了托管包装。托管包装使用户可以在 Windows 窗体客户端应用程序中显示网页。使用 WebBrowser 控件，可以实现应用程序中的 Internet Explorer Web 浏览功能，还可以禁用默认的 Internet Explorer 功能，并将该控件用做简单的 HTML 文档查看器。此外，可以使用该控件将基于 DHTML 的用户界面元素添加到窗体中，还可以隐藏这些元素在 WebBrowser 控件中承载的事实。通过这种方法，可以将 Web 控件和 Windows 窗体控件无缝地整合到一个应用程序中。

属性

WebBrowser 控件的常用属性及说明如表 W.1 所示。

表 W.1 WebBrowser 控件的常用属性及说明

属　性	描　述
CanGoBack	获取一个值，该值指示导航历史记录中的上一页是否可用，如果可用，GoBack 方法才能成功
CanGoForward	获取一个值，该值指示导航历史记录中的下一页是否可用，如果可用，GoForward 方法才能成功
Document	获取一个 HtmlDocument，它表示当前显示在 WebBrowser 控件中的网页
DocumentStream	获取或设置一个流，该流包含显示在 WebBrowser 控件中的网页的内容
DocumentText	获取或设置显示在 WebBrowser 控件中的页的 HTML 内容
DocumentTitle	获取当前显示在 WebBrowser 控件中的文档的标题
DocumentType	获取当前显示在 WebBrowser 控件中的文档的类型
ScrollBarsEnabled	获取或设置一个值，该值指示是否在 WebBrowser 控件中显示滚动条
StatusText	获取 WebBrowser 控件的状态文本
Url	获取或设置当前文档的 URL
WebBrowserShortcutsEnabled	获取或设置一个值，该值指示 WebBrowser 控件中是否启用了键盘快捷键

更多资源请访问稀酷客(www.ckook.com)

方法

WebBrowser 控件的常用方法及说明如表 W.2 所示。

表 W.2 WebBrowser 控件的常用方法及说明

方　法	描　述
GoBack	如果导航历史记录中的上一页可用，则将 WebBrowser 控件导航到该页
GoForward	如果导航历史记录中的下一页可用，则将 WebBrowser 控件导航到该页
GoHome	将 WebBrowser 控件导航到当前用户的主页
GoSearch	将 WebBrowser 控件导航到当前用户的默认搜索页
Hide	对用户隐藏控件
Navigate	已重载。将指定位置的文档加载到 WebBrowser 控件中
Print	使用当前打印和页面设置打印当前显示在 WebBrowser 控件中的文档
Refresh	已重载。已重写。重新加载当前显示在 WebBrowser 控件中的文档
ResetText	将 Text 属性重置为其默认值
Select	已重载。激活控件
Show	向用户显示控件
ShowPageSetupDialog	打开 Internet Explorer“页面设置”对话框
ShowPrintDialog	打开 Internet Explorer 的“打印”对话框，但不设置页眉或页脚值
ShowPrintPreviewDialog	打开 Internet Explorer 的“打印预览”对话框
ShowPropertiesDialog	打开当前文档的 Internet Explorer“属性”对话框
ShowSaveAsDialog	打开 Internet Explorer 的“保存网页”对话框，如果承载的文档不是 HTML 页，则打开其“保存”对话框
Stop	取消所有挂起的导航并停止所有动态页元素（如背景声音和动画）

事件

WebBrowser 控件的常用事件及说明如表 W.3 所示。

表 W.3 WebBrowser 控件的常用事件及说明

事　件	描　述
Click	此控件不支持此事件
DocumentCompleted	在 WebBrowser 控件完成加载文档时发生
DocumentTitleChanged	在 DocumentTitle 属性值更改时发生
EncryptionLevelChanged	在 WebBrowser 控件导航到或离开使用了加密的网站时发生
FileDownload	在 WebBrowser 控件下载文件时发生
ProgressChanged	在 WebBrowser 控件已更新有关要导航到的文档的下载进度的信息时发生

更多资源请访问稀酷客(www.ckook.com)

典型应用

本实例用 WebBrowser 控件制作了一个网络浏览器。该浏览器可以实现 IE 浏览器的基本功能。效果如图 W.1 所示。

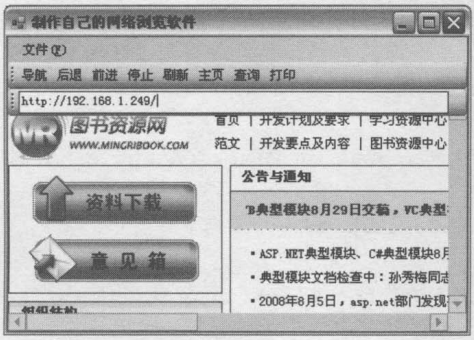


图 W.1　WebBrowser 控件的应用

主要代码如下：

```
    private void dToolStripMenuItem_Click(object sender, EventArgs e)
    {
        webBrowser1.ShowPageSetupDialog();//显示"页面设置"对话框
    }
    private void 打印浏览UToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        webBrowser1.ShowPrintPreviewDialog();//显示"打印预览"对话框
    }
    private void 打印PToolStripMenuItem_Click(object sender, EventArgs e)
    {
        webBrowser1.Print();//打印
    }
    private void 属性NToolStripMenuItem_Click(object sender, EventArgs e)
    {
        webBrowser1.ShowPropertiesDialog();//显示"属性"对话框
    }
    private void 退出IToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
    private void 保存DToolStripMenuItem_Click(object sender, EventArgs e)
    {
        webBrowser1.ShowSaveAsDialog();//显示"保存"对话框
    }
```


更多资源请访问稀酷客(www.ckook.com)

```
//文件导航
private void GoaButton_Click(object sender, EventArgs e)
{
    Navigate(toolStripTextBox1.Text);//文件导航
}
private void backButton_Click(object sender, EventArgs e)
{
    webBrowser1.GoBack();//如果导航记录可用
}
private void forwardButton_Click(object sender, EventArgs e)
{
    webBrowser1.GoForward();//导向下一页
}
private void stopButton_Click(object sender, EventArgs e)
{
    webBrowser1.Stop();//取消当前页
}
private void refreshButton_Click(object sender, EventArgs e)
{
    if (!webBrowser1.Url.Equals("about:blank"))
    {
        webBrowser1.Refresh();//使用控件重新加载页
    }
}
private void homeButton_Click(object sender, EventArgs e)
{
    webBrowser1.GoHome();//返回主页
}
private void searchButton_Click(object sender, EventArgs e)
{
    webBrowser1.GoSearch();//控件导航当前页
}
private void printButton_Click(object sender, EventArgs e)
{
    webBrowser1.Print();//打印
}
private void Navigate(String address) //导航方法
{
    if (String.IsNullOrEmpty(address)) return;
    if (address.Equals("about:blank")) return;
    if (!address.StartsWith("http://")) address = "http://" + address;
    try
    {
```


更多资源请访问稀酷客(www.ckook.com)

```
            webBrowser1.Navigate(new Uri(address));
        }
        catch (System.UriFormatException)
        {
            return;
        }
    }
    private void webBrowser1_Navigated(object sender,
    WebBrowserNavigatedEventArgs e)
    {
        toolStripTextBox1.Text = webBrowser1.Url.ToString();//Web 导航
后发生
    }
    private void webBrowser1_CanGoForwardChanged(object sender,
EventArgs e)
    {//CanGoForward 属性值更改时发生
        forwardButton.Enabled = webBrowser1.CanGoForward;
    }
    private void toolStripTextBox1_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)//如果按回车键
        {
            Navigate(toolStripTextBox1.Text);//调用自定义方法
        }
    }
```


WebClient 类

功能

提供向 URI 标识的资源发送数据和从 URI 标识的资源接收数据的公共方法。

命名空间

System.Net

属性

WebClient 类的常用属性及说明如表 W.4 所示。

更多资源请访问稀酷客(www.ckook.com)

表 W.4 WebClient 类的常用属性及说明

属　性	描　述
BaseAddress	获取或设置 WebClient 发出请求的基 URI
CachePolicy	对于此 WebClient 实例使用 WebRequest 对象获得的任何资源，获取或设置应用程序的缓存策略
Encoding	获取和设置用于上载和下载字符串的 Encoding
Headers	获取或设置与请求关联的标头名称/值对集合
IsBusy	了解是否存在进行中的 Web 请求
Proxy	获取或设置此 WebClient 对象使用的代理
QueryString	获取或设置与请求关联的查询名称/值对集合
ResponseHeaders	获取与响应关联的标头名称/值对集合
UseDefaultCredentials	获取或设置 Boolean 值，该值控制 DefaultCredentials 是否随请求一起发送



方法

WebClient 类的常用方法及说明如表 W.5 所示。

表 W.5 WebClient 类的常用方法及说明

方　法	描　述
CancelAsync	取消一个挂起的异步操作
Dispose	已重载。释放由 Component 占用的资源
DownloadData	已重载。以 Byte 数组形式通过指定的 URI 下载
DownloadDataAsync	已重载。以 Byte 数组形式下载指定的资源。这些方法不会阻止调用线程
DownloadFile	已重载。将具有指定 URI 的资源下载到本地文件
DownloadFileAsync	已重载。将指定的资源下载到本地文件。这些方法不会阻止调用线程
DownloadString	已重载。以 String 或 Uri 形式下载指定的资源
DownloadStringAsync	已重载。下载以 String 或 Uri 形式指定的资源。这些方法不会阻止调用线程
OpenRead	已重载。为从具有指定 URI 的资源下载的数据打开一个可读的流
OpenReadAsync	已重载。打开包含指定资源的可读流。这些方法不会阻止调用线程
OpenWrite	已重载。打开一个流以将数据写入具有指定 URI 的资源
OpenWriteAsync	已重载。打开一个流以将数据写入指定的资源。这些方法不会阻止调用线程
UploadData	已重载。将数据缓冲区上载到具有指定 URI 的资源
UploadDataAsync	已重载。将数据缓冲区上载到由 URI 标识的资源。此方法不会阻止调用线程
UploadFile	已重载。将本地文件上载到具有指定 URI 的资源

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
UploadFileAsync	已重载。将指定的本地文件上载到指定的资源。这些方法不会阻止调用线程
UploadString	已重载。将指定的字符串上载到指定的资源
UploadStringAsync	已重载。将指定的字符串上载到指定的资源。这些方法不会阻止调用线程
UploadValues	已重载。将名称/值集合上载到具有指定 URI 的资源
UploadValuesAsync	已重载。将指定的名称/值集合上载到指定的 URI 所标识的资源。这些方法不会阻止调用线程

事件

WebClient 类的常用事件及说明如表 W.6 所示。

表 W.6　WebClient 类的常用事件及说明

事　件	描　述
DownloadDataCompleted	在异步数据下载操作完成时发生
DownloadFileCompleted	在异步文件下载操作完成时发生
DownloadProgressChanged	在异步下载操作成功转换部分或全部数据后发生
DownloadStringCompleted	在异步资源下载操作完成时发生
OpenReadCompleted	在以下异步操作完成时发生，该操作打开包含某个资源的流
OpenWriteCompleted	在打开流以将数据写入资源的异步操作完成时发生
UploadDataCompleted	在异步数据上载操作完成时发生
UploadFileCompleted	在异步文件上载操作完成时发生
UploadProgressChanged	在异步上载操作成功转换部分或全部数据后发生
UploadStringCompleted	在异步字符串上载操作完成时发生
UploadValuesCompleted	在名称/值集合的异步上载完成时发生

While 语句

while 语句执行一个语句或语句块，直到指定的表达式计算为 False。

语法

```
while(表达式)
{
  循环体
}
```


更多资源请访问稀酷客(www.ckook.com)

示例

本示例用 while 语句循环输出 5 个数字。

```
static void Main(string[] args)
{
    Console.WriteLine("输出结果为：");
    int n = 1;
    while (n < 6)
    {
        Console.WriteLine("循环第{0} 次", n);
        n++;
    }
    Console.ReadLine();
}
```

运行结果如图 W.2 所示。

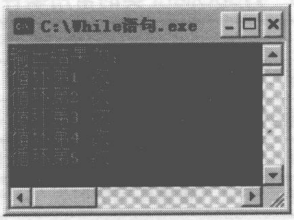


图 W.2　while 语句的应用

Width 属性

获取或设置控件的宽度。

语法

```
public int Width { get; set; }
```

属性值：控件的宽度（以像素为单位）。

WindowState 属性

获取或设置窗体的窗口状态。

语法

```
public FormWindowState WindowState { get; set; }
```


更多资源请访问稀酷客(www.ckook.com)

属性值：FormWindowState，表示窗体的窗口状态。默认为 FormWindowState.Normal。FormWindowState 枚举的各成员及说明如表 W.7 所示。

表 W.7　Url 属性的返回值

成　员	描　述
Maximized	最大化的窗口
Minimized	最小化的窗口
Normal	默认大小的窗口

WndProc 方法

处理 Windows 消息。

语法

```
protected virtual void WndProc (ref Message m)
```

WndProc 方法语法中的参数说明如表 W.8 所示。

表 W.8　WndProc 方法语法中的参数说明

参　数	描　述
m	要处理的 Windows 的 Message 信息

Write 方法

语法

- BinaryWriter 类的 Write 方法

```
public virtual void Write (byte value)
```

说明

此方法将一个无符号字节写入当前流，并将流的位置提升 1 个字节。

Write 方法的参数及说明如表 W.9 所示。

表 W.9　Write 方法的参数及说明

参　数	描　述
value	要写入的无符号字节

- CryptoStream 类的 Write 方法

更多资源请访问稀酷客(www.ckook.com)

```
public override void Write (byte[] buffer,int offset,int count)
```



说明

此方法将一个字节序列写入当前 CryptoStream，并将流中的当前位置提升写入的字节数。

Write 方法的参数及说明如表 W.10 所示。

表 W.10 Write 方法的参数及说明

参　数	描　述
buffer	字节数组。此方法将 count 个字节从 buffer 复制到当前流
offset	buffer 中的字节偏移量，从此偏移量开始将字节复制到当前流
count	要写入当前流的字节数

- FileStream 类的 Write 方法

```
public override void Write (byte[] array,int offset,int count)
```



说明

使用从缓冲区读取的数据将字节块写入该流。

Write 方法的参数及说明如表 W.11 所示。

表 W.11 Write 方法的参数及说明

参　数	描　述
array	指包含要写入该流的数据的缓冲区
offset	指数组中的从零开始的字节偏移量，从此处开始将字节复制到当前流
count	指要写入当前流的最大字节数

- StreamWriter 类的 Write 方法

```
public override void Write (char value)
```



说明

此方法将字符写入流。

Write 方法的参数及说明如表 W.12 所示。

表 W.12 Write 方法的参数及说明

参　数	描　述
value	要写入文本流中的字符

更多资源请访问稀酷客(www.ckook.com)

典型应用

本实例用 StreamWriter 类的 Write 方法和 CryptoStream 类的 Write 方法对指定的文件进行加密和解密的操作。运行结果如图 W.3 所示。



图 W.3　文件的加密与解密

文件加密的主要代码如下：

```
    private void button5_Click(object sender, EventArgs e)
    {
        if (textBox1.Text == "")
        { MessageBox.Show("请选择要加密的文件"); }
        else
        {
            try
            {
                string strPath = textBox1.Text;//加密文件的路径
                int intLent = strPath.LastIndexOf("\\") + 1;
                int intLong = strPath.Length;
                   //要加密的文件名称
                string strName = strPath.Substring(intLent, intLong -
intLent);
                int intTxt = strName.LastIndexOf(".");
                int intTextLeng = strName.Length;
                   //取出文件的扩展名
                string strTxt = strName.Substring(intTxt, intTextLeng -
intTxt);
                strName = strName.Substring(0, intTxt);
                //加密后的文件名及路径
                string strOutName = strPath.Substring(0, strPath.
LastIndexOf("\\")
     + 1) + strName + "Out" + strTxt;
                byte[] key = { 24, 55, 102, 24, 98, 26, 67, 29, 84, 19,
37, 118, 104,
    85, 121, 27, 93, 86, 24, 55, 102, 24, 98, 26, 67, 29, 9, 2, 49, 69,
73, 92 };
                byte[] IV ={ 22, 56, 82, 77, 84, 31, 74, 24, 55, 102, 24,
98, 26, 67,
```


更多资源请访问稀酷客(www.ckook.com)

```
29, 99 };
            RijndaelManaged myRijndael = new RijndaelManaged();
            FileStream fsOut = File.Open(strOutName, FileMode.Create,
FileAccess.Write);
            FileStream fsIn = File.Open(strPath, FileMode.Open,
FileAccess.Read);
            //写入加密文本文件
            CryptoStream csDecrypt = new CryptoStream(fsOut,
myRijndael.CreateEncryptor(key, IV), CryptoStreamMode.Write);
            //读加密文本
            BinaryReader br = new BinaryReader(fsIn);
            csDecrypt.Write(br.ReadBytes((int)fsIn.Length), 0,
(int)fsIn.Length);
            csDecrypt.FlushFinalBlock();
            csDecrypt.Close();
            fsIn.Close();
            fsOut.Close();
            if (MessageBox.Show(strOutName, "提示:加密成功!加密后的文
件名及路径为:\n" + "是否册除源文件", MessageBoxButtons.YesNo) ==
DialogResult.Yes)
            {
                File.Delete(strPath);
                textBox1.Text = "";
            }
            else
            { textBox1.Text = ""; }
        }
        catch (Exception ee)
        {
            MessageBox.Show(ee.Message);
        }
    }
}
```

文件解密的主要代码如下：

```
private void button4_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "")
    {
        MessageBox.Show("请选择要解密的文件路径");
    }
    else
    {
```


更多资源请访问稀酷客(www.ckook.com)

```
            string strPath = textBox1.Text;//加密文件的路径
            int intLent = strPath.LastIndexOf("\\") + 1;
            int intLong = strPath.Length;
              //要加密的文件名称
            string strName = strPath.Substring(intLent, intLong -
intLent);
            int intTxt = strName.LastIndexOf(".");
            int intTextLeng = strName.Length;
            strName = strName.Substring(0, intTxt);
            if (strName.LastIndexOf("Out") != -1)
            {
                strName = strName.Substring(0, strName.LastIndexOf("Out"));
            }
            else
            {
                strName = strName + "In";
            }
            //加密后的文件名及路径
            string strInName = strPath.Substring(0, strPath.LastIndexOf
("\\") + 1) +
    strName + ".txt";
            byte[] key = { 24, 55, 102, 24, 98, 26, 67, 29, 84, 19, 37,
118, 104, 85,
    121, 27, 93, 86, 24, 55, 102, 24, 98, 26, 67, 29, 9, 2, 49, 69, 73,
92 };
            byte[] IV ={ 22, 56, 82, 77, 84, 31, 74, 24, 55, 102, 24, 98,
26, 67, 29,
    99 };
            RijndaelManaged myRijndael = new RijndaelManaged();
            FileStream fsOut = File.Open(strPath, FileMode.Open,
    FileAccess.Read);
            CryptoStream csDecrypt = new CryptoStream(fsOut,
    myRijndael.CreateDecryptor(key, IV), CryptoStreamMode.Read);
            StreamReader sr = new StreamReader(csDecrypt);//把文件读出来
              //解密后文件写入一个新的文件
            StreamWriter sw = new StreamWriter(strInName);
            sw.Write(sr.ReadToEnd());
            sw.Flush();
            sw.Close();
            sr.Close();
            fsOut.Close();
            if (MessageBox.Show(strInName, "提示:解密成功!解密后的文件名及
    路径为:" + "是否删除源文件", MessageBoxButtons.YesNo) ==
```


更多资源请访问稀酷客(www.ckook.com)

```
    DialogResult.Yes)
            {
                File.Delete(strPath);
                textBox1.Text = "";
            }
            else
            {
                textBox1.Text = "";
            }
        }
    }
```

WriteLine 方法



语法

- Console 类的 WriteLine 方法

```
public static void WriteLine()
```



说明

将当前行结束符写入标准输出流。

```
public static void WriteLine(bool value)
```



说明

将指定布尔值的文本表示形式（后跟当前行结束符）写入标准输出流。

```
public static void WriteLine(char value)
```



说明

将指定的 Unicode 字符值（后跟当前行结束符）写入标准输出流。

```
public static void WriteLine(char[] buffer)
```


更多资源请访问稀酷客(www.ckook.com)



说明

将指定的 Unicode 字符数组（后跟当前行结束符）写入标准输出流。

```
public static void WriteLine(decimal value)
```



说明

将指定的 Decimal 值的文本表示形式（后跟当前行结束符）写入标准输出流。

```
public static void WriteLine(string value)
```



说明

将指定的字符串值（后跟当前行结束符）写入标准输出流。

```
public static void WriteLine(string format,params Object[] arg)
```



说明

使用指定的格式信息，将指定的对象数组（后跟当前行终止符）的文本表示形式写入标准输出流。

```
[CLSCompliantAttribute(false)]
public static void WriteLine(uint value)
```



说明

将指定的 32 位无符号的整数值的文本表示（后跟当前行的结束符）写入标准输出流。

WriteLine 方法的参数及说明如表 W.13 所示。

表 W.13　WriteLine 方法的参数及说明

参　数	描　述
value	要写入的值

更多资源请访问稀酷客(www.ckook.com)

续表

参　数	描　述
Buffer	Unicode 字符数组
format	格式字符串
arg	要使用 format 写入的对象的数组

- StreamWriter 类的 WriteLine 方法

```
public virtual void WriteLine ()
```



说明

将行结束符写入文本流。

返回值：默认行结束符是后跟换行符的回车符（“\r\n”），但使用 NewLine 属性可以更改此值。

WriteXml 方法

从 DataSet 写 XML 数据，还可以选择写架构。



语法

```
public void WriteXml(Stream stream)
```

WriteXml 方法语法中的参数说明如表 W.14 所示。

表 W.14　WriteXml 方法语法中的参数说明

参　数	描　述
stream	用于写入文件的 Stream 对象

更多资源请访问稀酷客(www.ckook.com)

XML

功能

XML 是 Extensible Markup Language 的缩写，它是一种类似于 HTML 的标记语言。XML 以简易而标准的方式保存各种信息（如文字和数字等信息），它适用于不同应用程序间的数据交换，而这种交换不以预先定义的一组数据结构为前提，增强了可扩展性。

结构

XML 是一种可以定义其他标识语言的元标识语言，在 XML 文档中，可以自定义标记和文档结构。

XML 文档总体上包括两部分：序言和文档元素。

- 序言

序言中包含 XML 声明、处理指令和注释。序言必须出现在 XML 文档的开始处。

上面示例代码中的第一行是 XML 声明，用于说明这是一个 XML 文档，并且给出版本号；第二行为处理指令，引用处理指令的目的是提供有关 XML 应用程序的信息，本示例中的处理指令表明该 XML 文件使用 CSS 样式表文件 style.css；第 3 行为注释语句。

- 文档元素

XML 文档中的元素是以树型分层结构排列的，元素可以嵌套在其他元素中。文档中只能有一个顶层元素，称为文档元素或者根元素，类似于 HTML 页中的 BODY 元素，其他所有元素都嵌套在文档元素中。文档元素中包含各种元素、属性、文本内容、字符、实体引用和 CDATA 区等。

在上面的示例代码中，文档元素是 RESUME，其起始和结束标记分别为<RESUME>、</RESUME>。在文档元素中定义了标记<NAME>、<EMAIL>、<HOMEPAGE>、<PUBLICATION>，它们分别表示公司名称、电子邮箱、主页地址和出版物信息；在标记<PUBLICATION>中定义了两个<BOOK>标记，用于说明出版物情况；<BOOK>标记中定义了标记<TITLE>和<PAGES>，它们分别表示出版物的名称和页码信息。

在 XML 文档中，元素指出了文档的逻辑结构，并且包含了文档的信息。

更多资源请访问稀酷客(www.ckook.com)

XML 在形式上与 HTML 相似，但实际上和 HTML 有很大区别。下面介绍 XML 和 HTML 的主要不同点：

（1）XML 是关于如何描述信息的语言，使用 XML 可以让应用程序知道哪些是数据，以及如何存储、组织和交换数据，而 HTML 是关于如何显示信息的语言。在 XML 中，标记仅用于描述数据本身，而在 HTML 中，浏览器通过编译 HTML 标记来显示数据。

（2）XML 是可扩展的标记语言，而在 HTML 中，所有的标记和文档结构都是预定义的。HTML 只允许使用当前 HTML 标准所支持的标记，而 XML 允许自定义标记和文档结构，其标记的定义是无限制的。

（3）XML 文档中的元素所表现的内容不能在浏览器中直接显示，而 HTML 文档中的元素可以在浏览器中直接显示。



语法要求

标准 XML 文档的语法要求如下：

（1）XML 文档必须有一个顶层元素，即文档元素，所有其他元素必须嵌入在文档元素中。

（2）元素嵌套要正确，即如果一个元素在另一个元素中开始，那么必须在同一元素中结束。

（3）每一个元素必须同时拥有起始标记和结束标记。

（4）起始标记中的元素类型名必须与相应结束标记中的名称完全匹配。

（5）元素类型名区分大小写。例如分别定义起始标记<TITLE>、结束标记</Title>，起始标记的类型名与结束标记的类型名不匹配，则元素是非法的。

（6）元素类型名称中可以包含字母、数字以及其他字母元素类型，也可以使用非英文字符，但名称不能以数字或者符号“-”开头，另外，名称中不能包含空格和冒号“:”。



命名空间

XML 文档中，定义的元素名是不固定的，因此当嵌套的两个 XML 文档使用相同名字描述两个不同类型的元素时，会发生冲突。XML Namespaces 命名空间提供了一种避免元素名冲突的方法。

Namespaces 命名空间是由 W3C 组织推荐提供的一种统一命名 XML 文档中元素和属性的机制。Namespaces 命名空间给每个元素添加一个独一无二的标志，在 XML 文件中使用定义的元素时，应该将元素的标志、元素名和属性连在一起使用，即指明元素的出处，这样就不会和其他同名元素混淆。

更多资源请访问稀酷客(www.ckook.com)

Namespaces 命名空间通过给标记名称添加一个 URL 网址来区别名称相同的标记，该命名空间需要在 XML 文档的开头部分声明，声明语法如下：

```
xmlns[:itemname]="URL"
```

XML 命名空间语法的参数及说明如表 X.1 所示。

表 X.1　XML 命名空间语法的参数及说明

参　数	描　述
xmlns	表示定义 Namespaces 命名空间
itemname	自定义的 Namespaces 命名空间前缀
URL	指定的网址，保证命名空间的唯一性

XmlReader 类

提供对 XML 数据进行快速、非缓存、只进访问的读取器。

命名空间

System.Xml

属性

XmlReader 类的常用属性及说明如表 X.2 所示。

表 X.2　XmlReader 类的常用属性及说明

属　性	描　述
BaseURI	当在派生类中被重写时，获取当前节点的基 URI
CanReadBinaryContent	获取一个值，该值指示 XmlReader 是否实现二进制内容读取方法
CanReadValueChunk	获取一个值，该值指示 XmlReader 是否实现 ReadValueChunk 方法
Depth	当在派生类中被重写时，获取 XML 文档中当前节点的深度
EOF	当在派生类中被重写时，获取一个值，该值指示此读取器是否定位在流的结尾
HasValue	当在派生类中被重写时，获取一个值，该值指示当前节点是否可以具有 Value
LocalName	当在派生类中被重写时，获取当前节点的本地名称
NamespaceURI	当在派生类中被重写时，获取读取器定位在其上的节点的命名空间 URI（采用 W3C 命名空间规范中定义的形式）
NameTable	当在派生类中被重写时，获取与该实现关联的 XmlNameTable
NodeType	当在派生类中被重写时，获取当前节点的类型
ReadState	当在派生类中被重写时，获取读取器的状态
SchemaInfo	获取作为架构验证结果分配给当前节点的架构信息

更多资源请访问稀酷客(www.ckook.com)

续表

属　性	描　述
Settings	获取用于创建此 XmlReader 实例的 XmlReaderSettings 对象
Value	当在派生类中被重写时，获取当前节点的文本值
ValueType	获取当前节点的公共语言运行库（CLR）类型
XmlLang	当在派生类中被重写时，获取当前的 xml:lang 范围
XmlSpace	当在派生类中被重写时，获取当前的 xml:space 范围



方法

XmlReader 类的常用方法及说明如表 X.3 所示。

表 X.3　XmlReader 类的常用方法及说明

方　法	描　述
Close	当在派生类中被重写时，将 ReadState 更改为 Closed
Read	当在派生类中被重写时，从流中读取下一个节点
ReadAttributeValue	当在派生类中被重写时，将属性值解析为一个或多个 Text、EntityReference 或 EndEntity 节点
ReadContentAs	将内容作为指定类型的对象读取
ReadContentAsBase64	读取内容并返回 Base64 解码的二进制字节
ReadContentAsBinHex	读取内容并返回 BinHex 解码的二进制字节
ReadContentAsBoolean	将当前位置的文本内容作为 Boolean 读取
ReadContentAsDateTime	将当前位置的文本内容作为 DateTime 对象读取
ReadContentAsDecimal	将当前位置的文本内容作为 Decimal 对象读取
ReadContentAsDouble	将当前位置的文本内容作为双精度浮点数读取
ReadContentAsFloat	将当前位置的文本内容作为单精度浮点数读取
ReadContentAsInt	将当前位置的文本内容作为 32 位有符号整数读取
ReadContentAsLong	将当前位置的文本内容作为 64 位有符号整数读取
ReadContentAsObject	将当前位置的文本内容作为 Object 读取
ReadContentAsString	将当前位置的文本内容作为 String 对象读取
ReadElementContentAs	已重载。读取当前元素，并将内容作为指定类型的对象返回
ReadElementString	已重载。这是一个用于读取简单纯文本元素的 Helper 方法
ReadEndElement	检查当前内容节点是否为结束标记并将读取器推进到下一个节点
ReadInnerXml	当在派生类中被重写时，将所有内容（包括标记）当做字符串读取
ReadOuterXml	当在派生类中被重写时，读取表示该节点和所有它的子级的内容（包括标记）
ReadStartElement	已重载。检查当前节点是否为元素并将读取器推进到下一个节点
ReadString	当在派生类中被重写时，将元素或文本节点的内容当做字符串读取
ReadSubtree	返回新的 XmlReader 实例，此实例可用于读取当前节点及其所有子节点

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
ReadToDescendant	已重载。让 XmlReader 前进到下一个匹配的子代元素
ReadToFollowing	已重载。一直读取，直到找到命名元素
ReadToNextSibling	已重载。让 XmlReader 前进到下一个匹配的同级元素
ReadValueChunk	读取嵌入在 XML 文档中的大量文本流
Skip	跳过当前节点的子级

XmlWriter 类

一个编写器，该编写器提供一种快速、非缓存和只进的方式来生成包含 XML 数据的流或文件。



命名空间

System.Xml



属性

XmlWriter 类的常用属性及说明如表 X.4 所示。

表 X.4　XmlWriter 类的常用属性及说明

属　性	描　述
Settings	获取用于创建此 XmlWriter 实例的 XmlWriterSettings 对象
WriteState	当在派生类中被重写时，获取编写器的状态
XmlLang	当在派生类中被重写时，获取当前的 xml:lang 范围
XmlSpace	当在派生类中被重写时，获取表示当前 xml:space 范围的 XmlSpace



方法

XmlWriter 类的常用方法及说明如表 X.5 所示。

表 X.5　XmlWriter 类的常用方法及说明

方　法	描　述
Close	当在派生类中被重写时，关闭此流和基础流
LookupPrefix	当在派生类中被重写时，返回在当前命名空间范围中为该命名空间 URI 定义的最近的前缀
WriteAttributes	当在派生类中被重写时，写出在 XmlReader 中当前位置找到的所有属性
WriteAttributeString	已重载。当在派生类中被重写时，写出具有指定值的属性

更多资源请访问稀酷客(www.ckook.com)

续表

方　法	描　述
WriteBase64	当在派生类中被重写时，将指定的二进制字节编码为 Base64 并写出结果文本
WriteBinHex	当在派生类中被重写时，将指定的二进制字节编码为 BinHex 并写出结果文本
WriteCData	当在派生类中被重写时，写出包含指定文本的<![CDATA[...]]>块
WriteCharEntity	当在派生类中被重写时，为指定的 Unicode 字符值强制生成字符实体
WriteChars	当在派生类中被重写时，以每次一个缓冲区的方式写入文本
WriteComment	当在派生类中被重写时，写出包含指定文本的注释<!--...-->
WriteNode	已重载。将所有内容从源对象复制到当前编写器实例
WriteProcessingInstruction	当在派生类中被重写时，写出在名称和文本之间带有空格的处理指令，如：<?name text?>
WriteQualifiedName	当在派生类中被重写时，写出命名空间限定的名称。此方法查找位于给定命名空间范围内的前缀
WriteRaw	已重载。当在派生类中被重写时，手动书写原始标记
WriteStartAttribute	已重载。当在派生类中被重写时，书写属性的起始内容
WriteStartDocument	已重载。当在派生类中被重写时，书写 XML 声明
WriteStartElement	已重载。当在派生类中被重写时，写出指定的开始标记
WriteString	当在派生类中被重写时，书写给定的文本内容
WriteSurrogateCharEntity	当在派生类中被重写时，为代理项字符对生成并书写代理项字符实体
WriteValue	已重载。编写单一的简单类型化值
WriteWhitespace	当在派生类中被重写时，写出给定的空白

更多资源请访问稀酷客(www.ckook.com)

YEAR 函数

返回表示指定日期中的年份的整数。

语法

```
YEAR (date)
```

返回值：Int 类型，4 位数的年份。

YEAR 函数的参数及说明如表 Y.1 所示。

表 Y.1　YEAR 函数的参数及说明

参　数	描　述
date	datetime 或 smalldatetime 类型的表达式

示例

本示例用 YEAR 函数返回 mm-dd-yyyy 格式日期中的年份。

```
SELECT YEAR('11/23/2006')
```

返回结果为：2006

本示例用 YEAR 函数返回 yyyy-mm-dd 格式日期中的年份。

```
SELECT YEAR('2006-11-23')
```

返回结果为：2006

更多资源请访问稀酷客(www.ckook.com)

Z

Zoom 属性

获取或设置一个值，该值指示页面的显示大小。



语法

```
public double Zoom { get; set; }
```

属性值：指示页面显示大小的值。值为 1.0 指示实际大小。



示例

本示例用 Zoom 属性设置页面的显示大小。

```
this.PrintPreviewControl1.Zoom = 0.25;
```


更多资源请访问稀酷客(www.ckook.com)